AD-A045 444 MICHIGAN UNIV ANN ARBOR DEPT OF NAVAL ARCHITECTURE --ETC F/G 9/2
SHIPSIM/OPTSIM: SIMULATION PROGRAM FOR STATIONARY LINEAR OPTIMA--ETC(U)
JUN 77 M G PARSONS, J E GREENBLATT N00014-76-C-0751

UNCLASSIFIED 188 ONR-CR215-249-1 NL

1 OF 2
AD
A045444

AD A045444

REPORT ONR-CR215-249-1

(12)

OFFICE OF NAVAL RESEARCH

ONR

USN

SHIPSIM/OPTSIM: SIMULATION PROGRAM FOR STATIONARY LINEAR OPTIMAL STOCHASTIC CONTROL SYSTEMS

MICHAEL G. PARSONS
JEFFREY E. GREENBLATT

DEPARTMENT OF NAVAL ARCHITECTURE AND MARINE ENGINEERING
THE UNIVERSITY OF MICHIGAN
ANN ARBOR, MICHIGAN 48109

CONTRACT N00014-76-C-0751

23 JUNE 1977

INTERIM REPORT FOR PERIOD 1 MAY 76 - 1 JUNE 1977

PREPARED FOR THE

OFFICE OF NAVAL RESEARCH·800 N. QUINCY ST.·ARLINGTON·VA·22217

NOTICES

Change of Address

Organizations receiving reports on the initial distribution list should confirm correct address. This list is located at the end of the report. Any change of address or distribution should be conveyed to the Office of Naval Research, Code 211, Arlington, Virginia 22217.

Disposition

When this report is no longer needed, it may be transmitted to other authorized organizations, or properly destroyed. Do not return it to the originator or the monitoring office.

Disclaimer

The findings in this report are not to be construed as an official Department of Defense or Military Department position unless so designated by other official documents.

Reproduction

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: ONR-CR215-249-1	2. GOVT ACCESSION NO.
	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle): SHIPSIM/OPTSIM: Simulation Program for Stationary Linear Optimal Stochastic Control Systems.	5. TYPE OF REPORT & PERIOD COVERED: Interim Report 1 May 76- 1 June 77
	6. PERFORMING ORG. REPORT NUMBER: 188
7. AUTHOR(s): Michael G. Parsons, Jeffrey E. Greenblatt	8. CONTRACT OR GRANT NUMBER(s): N00014-76-C-0751
9. PERFORMING ORGANIZATION NAME AND ADDRESS: Department of Naval Architecture and Marine Engineering - North Campus- Univ. of Michigan, Ann Arbor, Michigan 48109	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS: Office of Naval Research, 800 North Quincy Street, Arlington, VA. 22217	12. REPORT DATE: 23 June 1977
	13. NUMBER OF PAGES: ii + 103 (12) 108p.
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)	15. SECURITY CLASS. (of this report): UNCLASSIFIED
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

401 188

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Simulation
Stochastic control
Optimal control

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The SHIPSIM/OPTSIM computer program for the simulation of stationary, linear optimal stochastic control systems is presented. This program consists of two separate entities:

- SHIPSIM is a general continuous system simulation program designed to provide a versatile simulation capability with simple user input. This program can be used for <u>any</u> continuous systems simulation problem which can be defined

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

in the three user-supplied subroutines, and

OPTSIM is a group of subroutines which are run under SHIPSIM for the simulation of the response of stationary, linear optimal stochastic control systems to initial condition errors and specific user-supplied process disturbances while subject to measurement white noise.

These two program groups are described; User's Documentation, Programmer's Documentation, and listings are included as appendices. A simulation of an optimal stochastic path controller for a tanker operating in shallow water is presented as an example. The programs are written in FORTRAN IV and with the exception of the plotting portion of SHIPSIM, they are essentially independent of the Michigan Terminal System (MTS).

No. 188
ONR Report ONR-CR215-249-1
June 1977

SHIPSIM/OPTSIM: Simulation Program for Stationary, Linear Optimal Stochastic Control Systems

Michael G. Parsons
Jeffrey E. Greenblatt

These programs were developed in support
of research carried out under
ONR Contract N00014-76-C-0751

Department of Naval Architecture
and Marine Engineering
College of Engineering
The University of Michigan
Ann Arbor, Michigan 48109

ABSTRACT

The SHIPSIM/OPTSIM computer program for the simulation of stationary, linear optimal stochastic control systems is presented. This program consists of two separate entities:

- SHIPSIM is a general continuous system simulation program designed to provide a versatile simulation capability with simple user input. This program can be used for any continuous systems simulation problem which can be defined in the three user-supplied subroutines.
- OPTSIM is a group of subroutines which are run under SHIPSIM for the simulation of the response of stationary, linear optimal stochastic control systems to initial condition errors and specific user-supplied process disturbances while subject to measurement white noise.

These two program groups are described; User's Documentation, Programmer's Documentation, and listings are included as appendices. A simulation of an optimal stochastic path controller for a tanker operating in shallow water is presented as an example. The programs are written in FORTRAN IV and with the exception of the plotting portion of SHIPSIM, they are essentially independent of the Michigan Terminal System (MTS).

TABLE OF CONTENT

I. INTRODUCTION

The computer programs presented in this report have been developed in support of research performed on the optimal stochastic path control of surface ships in shallow water.[1] The programs were written to have general utility and with the documentation provided here should be of value to others. The SHIPSIM program has already been used in other research and teaching applications in the Department of Naval Architecture and Marine Engineering. The SHIPSIM/OPTSIM program has been written to be used in conjunction with the Department's OPTSYS program which can very quickly and cheaply provide the design of the optimal controller and Kalman-Bucy filter for a stationary, linear system subjected to white noise process disturbances and measurement noise. The OPTSYS program was originally written by W. Earl Hall, Jr.[2] at Stanford University; a version of this program has been utilized in our research on the path control of ships. The OPTSYS program can evaluate the Root Mean Square (RMS) response of the optimal system to the design disturbances and noise. It is also often desirable to simulate the response of the controlled system to other specific process disturbances and/or initial condition errors. The SHIPSIM/OPTSIM program allows this simulation with a minimum of additional effort and expense.

To provide the maximum utility, the SHIPSIM and SHIPSIM/OPTSIM programs have been written to be completely separate. Users without an interest in stochastic control systems can use SHIPSIM alone for any continuous system simulation problem which can be defined in the user-supplied subroutines. This report has, therefore, been written so that a reader interested only in SHIPSIM need only to consult Section III "Description of SHIPSIM" and Appendices A,D, and G which contain the SHIPSIM documentation and listing. For the general reader, Section II contains a brief review of the stochastic control of linear systems, Section IV contains a description of OPTSIM as run under SHIPSIM, and Section V contains a short example use of the SHIPSIM/OPTSIM program for the simulation of the stochastic path control of a tanker. Appendices B, E, and G include the documentation and listing for OPTSIM. Appendix C includes the User's Documentation for our version of the OPTSYS program for additional reference. The documentation Appendices have been written to be essentially self-contained so they can be reproduced separately for subsequence use with the programs.

II. OPTIMAL CONTROL OF LINEAR STOCHASTIC SYSTEMS

Many physical control problems can most realistically be represented using stochastic disturbances and measurement noise. The response of such a system may then also be treated as a stochastic quantity. The optimal control of linear stochastic systems[3,4,5] will be reviewed very briefly here as an introduction to the description of the SHIPSIM/OPTSIM program which follows. The engineering approach can be to model stochastic physical systems as Gauss-Markov processes which can be represented by the state vector of a linear dynamical system forced by a gaussian purely-random process where the initial state vector is also gaussian or normally distributed. Thus, we can represent the system by the following:

$$\overset{n\times 1}{\dot{\underline{x}}} = F\underline{x} + G\overset{m\times 1}{\underline{u}} + \Gamma\overset{q\times 1}{\underline{w}} \quad , \tag{1}$$

where the open-loop dynamics matrix F, the control distribution matrix G, and the disturbance distribution matrix Γ are assumed stationary (constant) here. The condition of the system is completely represented by the mean value n-vector $\overline{\underline{x}}$ and covariance matrix X for the n state (differentiated) variables; i.e.,

$$\overline{\underline{x}}(t) \equiv E[\underline{x}(t)] \quad \text{with} \quad \overline{\underline{x}}(t_o) = \overline{\underline{x}}_o \ , \tag{2}$$

$$X(t) \equiv E[(\underline{x}(t)-\overline{\underline{x}}(t))(\underline{x}(t)-\overline{\underline{x}}(t))^T] \quad \text{with} \quad X(t_o)=X_o \quad , \tag{3}$$

where E[...] is the expected value or ensemble average over the many possible observations at time t. The m non-differentiated variables in $\underline{u}$ are the control variables. The q variables in $\underline{w}$ are the process disturbances which are gaussian purely-random processes or white noise. White noise is an idealized, very-jittery process which can be viewed as the limit of a sequence of impulses with random magnitude and random time of occurence. The impulses average zero over time but have an average square magnitude given by $\sigma(t)$ squared. We thus have,

$$E[\underline{w}(t)] = 0 \ , \tag{4}$$

$$E[(\underline{w}(t)-\overline{\underline{w}}(t))(\underline{w}(\tau)-\overline{\underline{w}}(\tau))^T] = Q(t)\delta(t-\tau) \quad , \tag{5}$$

where Q is the power spectral density matrix and $\delta(t-\tau)$ is the Dirac delta function. It is also assumed that there is no correlation between the process disturbance and the initial condition of the system; i.e.

$$E[(\underline{w}(t)-\overline{\underline{w}}(t))(\underline{x}(t_o)-\overline{\underline{x}}_o)^T] = 0. \tag{6}$$

The control problem is to develop the optimal state variable feedback control,

$$\underline{u} = C\underline{x} \quad , \tag{7}$$

where C is the control feedback gain matrix. In general, not all of the needed states in $\underline{x}$ are readily measured. Further, it is not necessary to measure all the states if it is possible to estimate the remaining states from those which are most easily measured. In the stochastic case we may have p measurements available represented by,

$$\overset{px1}{\underline{z}} = H\underline{x} + \underline{v} \quad , \tag{8}$$

where H is the measurement distribution matrix (assumed constant here) and $\underline{v}$ is a vector of white measurement noise with statistical properties,

$$E[\underline{v}(t)] = 0 \quad , \tag{9}$$

$$E[\underline{v}(t)\underline{v}(\tau)^T] = R(t)\delta(t-\tau) \quad , \tag{10}$$

$$E[\underline{v}(t)\underline{w}(t)^T] = E[\underline{v}(t)(\underline{x}(t_o)-\overline{\underline{x}}_o)^T] = 0. \tag{11}$$

The matrix R is the power spectral density of the measurement noise. Equation (11) states that there is no correlation between the measurement noise and the process disturbance or the initial state of system. The elements of $\underline{z}$ may be measurements of specific states or linear combinations of the states.

The Separation Theorem[3] states that the optimal way to control the system eq. (1) using the information available in the noisy measurements eq. (8) is to design the controller gains eq. (7) neglecting $\underline{w}$ and $\underline{v}$ and thus assuming perfect knowledge of $\underline{x}$. The noisy measurements $\underline{z}$ can be utilized in an optimal stochastic observer (state estimator) or Kalman-Bucy filter to produce a maximum-likelihood estimate of the state $\hat{\underline{x}}$ and the optimal control will then be,

$$\underline{u} = C\hat{\underline{x}} \quad . \tag{12}$$

Thus, the controller design is completely separated from the processing of the noisy measurements to produce the best estimate of the current state of the system. An overall schematic of such a stochastic control system is shown in Figure 1. The measurements $\underline{z}$ from the sensors are used in the optimal stochastic observer to produce an estimate of the states $\hat{\underline{x}}$ which are then used in the optimal controller to produce the control signals $\underline{u}$ given to the actuators.

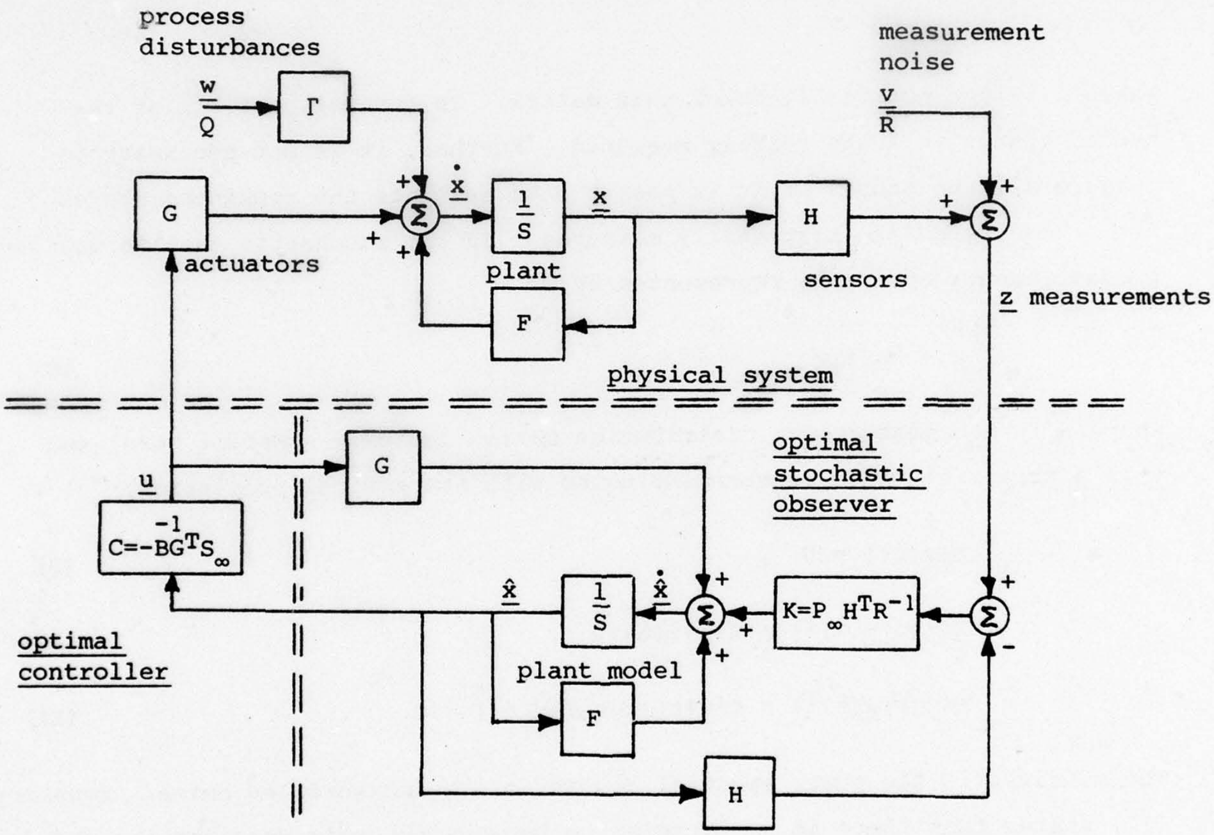


Figure 1. Overall Schematic of Optimal Stochastic Control System

An optimal control can be defined in many ways. The most common when we want to control $\underline{x}$ near zero using reasonable values of control $\underline{u}$ is to use the control which minimizes a linear quadratic cost function,

$$J = \frac{1}{2}\int_{t_o}^{t_f} (\underline{x}^T A\underline{x} + \underline{u}^T B\underline{u})\,dt \quad , \tag{13}$$

where the A and B matrices are initially established by the designer to reflect the relative weighting of errors in the various states and the use of the various controls. The A and B matrices are usually diagonal with at least some nonzero diagonal elements selected to be,

$$A_{ii} = \frac{1}{x_{oi}^2} \quad \text{and} \quad B_{jj} = \frac{1}{u_{oj}^2} \quad , \tag{14}$$

where u_{oj} is an acceptable amount of control j to be used when the state i deviates x_{oi} from zero. It is usually necessary to modify the weighting matrices A and B and iterate on the design based on the evaluated response of the controlled system. The calculus of variations[3,8] can be utilized to show that the control which minimizes eq. (13) for a stationary system is given by,

$$C = -B^{-1}G^{T}S_{\infty} \quad , \tag{15}$$

where S_{∞} is the steady-state solution of a matrix Riccati equation which is independent of $\underline{w}(Q)$ and $\underline{v}(R)$; i.e.,

$$\dot{S} = -SF-F^{T}S+SGB^{-1}G^{T}S-A. \tag{16}$$

An efficient way to obtain the steady-state solution, S_{∞}, is to utilize the technique of eigenvector decomposition first proposed by MacFarlane[6] and Potter.[7] This method was developed into a practical design computer program by Bryson and Hall[2] in Hall's OPTSYS program which uses the QR algorithm to solve the eigensystem. User's Documentation for the Michigan version of this program is included in Appendix C for information.

The second half of the design problem is to develop the optimal stochastic observer. Again, the calculus of variations[8] can be utilized to show that the maximum-likelihood estimate of the state is produced by the filter given by,

$$\dot{\hat{\underline{x}}} = F\hat{\underline{x}} + G\underline{u} + K(\underline{z}-H\hat{\underline{x}}) \;;\; \hat{\underline{x}}(t_o) = \bar{\underline{x}}_o \,, \tag{17}$$

where the filter feedback gain matrix K for a stationary system is given by,

$$K = P_{\infty}H^{T}R^{-1} \quad , \tag{18}$$

where P_∞ is the steady-state solution of the matrix Riccati equation,

$$\dot{P} = FP + PF^T + \Gamma Q \Gamma^T - PH^T R^{-1} HP \quad . \tag{19}$$

The matrix P is the covariance of the error of the estimate of the state; i.e.,

$$P(t) \equiv E[(\underline{\hat{x}}(t)-\underline{x}(t))(\underline{\hat{x}}(t)-\underline{x}(t))^T] \quad . \tag{20}$$

In eq. (17), it can be seen that the estimate $\underline{\hat{x}}$ is assumed to follow the same dynamics as $\underline{x}$ (excluding $\Gamma\underline{w}$) and that if the measurement which would result from $\underline{\hat{x}}$ (excluding $\underline{v}$) deviates from the actual measurement $\underline{z}$ a correction is introduced to drive $\underline{\hat{x}}$ closer to $\underline{x}$. In eq. (19), it can be seen that, as expected, large process disturbances (high Q) and large measurement noise (high R) will increase P, the error in the estimate produced by the stochastic observer. An efficient way to obtain the steady-state solution to eq. (19), P_∞, is again by eigenvector decomposition as implemented in the OPTSYS program described in Appendix C.

In modeling physical systems, it is not always realistic to assume that the process disturbance or measurement noise is white noise. If the process disturbance is a random quantity which changes very rapidly compared to the time response of the system, it is reasonable to assume the process disturbance is white noise as in Eq. (1). However, if the process disturbance is a random quantity which changes very slowly compared to the time response of the system (perhaps a tidal current effect on a passing ship), it is reasonable to assume the process disturbance to be a random bias or constant. This can be incorporated into the above treatment by defining an additional state variable or variables such that,

$$\dot{x}_{n+1} = 0 \ , \tag{21}$$

and $x_{n+1}(t_o)$ is random. This state variable is included in an augmented state vector of length n+1 and $\underline{w}$ might then be zero.

If the process disturbance is a random quantity which changes on about the same time scale as the response of the system, it must be modeled as something between white noise and a random bias. This is accomplished by the use of various shaping filters and again augmenting the state vector. The simplest shaping filter produces an exponentially correlated disturbance[3] by driving

a first-order system by white noise. A new state variable is defined as follows:

$$\tau_c \dot{x}_{n+1} + x_{n+1} = w \quad , \tag{22}$$

where τ_c is the correlation time and w is white noise. The state vector can then be augmented to an n+1 vector and the total system is still disturbed by white noise as in eq. (1). The shaping filter processes the white noise to produce a new disturbance x_{n+1} which is random but with a characteristic time constant τ_c of about the same order as the response time of the system. Other higher-order shaping filters are possible to model more complex disturbances.[3,5,9]

If a physical system, its disturbances, and the noise in its measurements are modeled as described above, a design program such as the OPTSYS program can be used to produce the optimal control gains C and the optimal filter gains K. The OPTSYS program can also produce the Root Mean Square (RMS) response of the system to the design disturbances as represented in the power spectral densities Q and R. The RMS response is, however, not that meaningful to many engineers. Also, specific physical disturbances will often be modeled through the use of shaping filters and the designer may want to know the response of the controlled system to the specific disturbances. Thus, there is often a need to simulate the response of the optimal stochastic control system to specific process disturbances and initial condition errors while subject to the measurement noise. The SHIPSIM/OPTSIM simulation program was developed to allow the simulation of these systems with a minimum of effort and computer programming. This simulation program is a valuable complement to the OPTSYS program for use in control system design.

III. Description of SHIPSIM

In developing a simulation of the optimal stochastic control systems for the path control of surface ships[1], it was decided to develop and document these programs so they could be of general use in the future. Further, it was decided to separate the input/output and numerical integration portions of the program from those portions specifically related to the stochastic control problem. This is the SHIPSIM-OPTSIM dividing point and makes SHIPSIM a general continuous systems simulation program which has a very wide range of applications and usefulness. It has already been used in a number of additional research and teaching applications within the Department of Naval Architecture and Marine Engineering at the University of Michigan.

The authors have research and teaching experience with continuous system simulations written from scratch using existing numerical integration subroutines[10] and with the use of higher-level, problem-oriented continuous systems simulation languages[11] specifically IBM's Continuous Systems Modeling Program (CSMP).[12] Based on this experience, it was felt that a valuable compromise between these two approaches would be a FORTRAN IV batch-type program with flexible input, output, and integration specification and control features to which the user would only have to add a few user-supplied subroutines which define his particular simulation problem in a standard form. This is similar to the modular approach taken in the development of our nonlinear, constrained, parameter optimization program[13] which has proven very useful in research and teaching applications. The user can work at the FORTRAN level without the need to learn a new, higher-level language but can still have available many of the labor saving features of a higher-level language such as CSMP. We have not conducted a specific test but SHIPSIM should also be significantly cheaper to run than CSMP since the CSMP-to-FORTRAN translation step is not needed.

The only computational function of SHIPSIM is to integrate a set of $n \leq 25$ coupled, nonlinear, first-order differential equations from an initial condition,

$$\underset{n\times 1}{\dot{\underline{Y}}(t)} = \underline{f}(\underline{Y}(t),t) \quad , \qquad \underline{Y}(t_o) = \underline{Y}o \quad , \tag{23}$$

and to perform a set of $m \leq 5$ nonlinear auxiliary calculations,

$$\underset{m\times 1}{\underline{z}(t)} = \underline{g}(\underline{Y}(t),t) \quad , \tag{24}$$

at each program output point. The output can be printed output of any selected or all elements of $\underline{Y}$ and all elements of $\underline{z}$ at specified points in t; CALCOMP plots can also be easily obtained for any selected elements of $\underline{Y}$ and $\underline{z}$ as functions of t. The user can select either a fixed step-size Euler (rectangular) integration or a variable step-size, guaranteed-error, fourth-order Kutta-Merson integration method.[14] The simulation can be performed in a series of sequential pre-specified integration segments using a different integration method and different integration and output specifications in each segment. The overall integration can be stopped based on the value of one of the elements of $\underline{Y}$ as well as on the value of the independent variable. Multiple runs can be made by changing the entire SHIPSIM and/or model input data sets for subsequent runs or by changing only specific values in the data sets used in the previous run. This substantial flexibility is achieved with fairly brief and simple SHIPSIM input data.

Euler or rectangular integration gives the value of $\underline{Y}$ at time $t+\Delta t$ as,

$$\underline{Y}(t+\Delta t) = \underline{Y}(t) + \underline{\dot{Y}}(t)*\Delta t \; ; \tag{25}$$

i.e., it evaluates the derivatives once at time t and assumes them constant over the integration step Δt. Euler integration is thus very simple and efficient. For processes with generally small and/or smooth variations in $\underline{\dot{Y}}$ it can yield acceptable answers when a suitably small integration step-size, Δt, is used. *When the step-size is too large it is possible for the integration results to completely diverge from the correct results.* Euler integration can also successfully handle discrete changes in $\underline{\dot{Y}}$ when Δt is kept small. If $\underline{\dot{Y}}$ experiences large and rapid changes in magnitude, the acceptable value of Δt may be so small that excessive CPU time will be needed to complete the integration. The integration step-size used in a region where $\underline{\dot{Y}}$ changes rapidly would be very wasteful if also used in regions where $\underline{\dot{Y}}$ changes more slowly. For this reason (and to facilitate variations in output), SHIPSIM allows a specific integration run to be specified with up to five integration segments each with a separate integration method and/or step-size. Euler integration has the disadvantage that the error of the integration results is unknown. It is therefore essential to perform test integration runs with a

reduced step-size to verify that the results are acceptably accurate.

Kutta-Merson integration[14,15] is much more complex than Euler integration but provides a dynamically varying integration step-size which is automatically doubled or halved as necessary to produce guaranteed, user-specified absolute and/or relative error in the results. This step-size will be short where $\underline{\dot{Y}}$ changes rapidly and much larger where $\underline{\dot{Y}}$ changes more slowly. There is much more computational overhead than with Euler integration but guaranteed error is provided and the integration step-size is never any shorter than necessary. Improved integration cost is thus possible in some cases. A major problem with Kutta-Merson integration is that it may be very difficult to meet the specified error limits (particularly at points where $\underline{\dot{Y}}$ is discontinuous in value or slope) without shortening the integration step-size to the point where excessive CPU time is used. To protect against excessive reduction in step-size without user interaction to relax error specifications, SHIPSIM includes capability to limit the number of times the initially specified step-size will be automatically halved (cut). If this number of cuts is exceeded, the integration is terminated and an error message is printed. Kutta-Merson integration is able to predict the integration error by evaluating the derivatives at five points in each integration step as compared to Euler integration which evaluates the derivatives only once for each integration step.

To define a particular problem for SHIPSIM, the user must provide a subroutine DERIV to evaluate the derivatives eq. (23) and a subroutine INPUT to provide the problem dependent input to the program. If the user also wants to output quantities in addition to the n integrated variables; i.e., if m>0, the user must also provide a subroutine ACALC to perform the auxiliary calculations eq. (24). If m=0, a file containing a dummy version of ACALC without executable code is available to be called in order to allow program loading without an error message. Files containing the object code (compiled) versions of these user-supplied subroutines must be concatenated with the SHIPSIM object file on the Michigan Terminal System (MTS) $RUN command to link the entire program together. The program can be easily run from the terminal using data files. Printed output can be taken on the terminal or line printer as specified. CALCOMP plots can be generated off-line using a SHIPSIM produced plotfile.

The macro structure of SHIPSIM is shown in Figure 2. The three user-supplied subroutines are shown across the bottom. It is important to emphasize how each of these subroutines are called by SHIPSIM since this establishes what the user may and should put in each subroutine. Subroutine INPUT is called once at the beginning of the simulation run. It must input or assign NEQ, the number of equations to be integrated by SHIPSIM (n in equation (23)). It must also obtain any problem dependent input and perform any problem input verification output and initialization needed by the user. The input must be transmitted to the remainder of the user-supplied subroutines by separate subroutine calls (as in OPTSIM which follows) or by labeled COMMON. The COMMON must be labeled since SHIPSIM already utilizes unlabeled COMMON. Subroutine DERIV is called at least once per integration step by whichever integration subroutine has been selected. It may thus be called hundreds or thousands of times in a simulation run. Subroutine ACALC is used to calculate the value of those non-integrated quantities which the user wishes to see in the printed or plotted output. It is thus called only at the specified print and plot points. Often these calculations (perhaps the value of a control) will duplicate those performed in DERIV but they should be repeated again in ACALC since the most recent value established in DERIV will be prior to the actual print or plot point of interest. The example in Appendix A User's Documentation for SHIPSIM illustrates such a case.

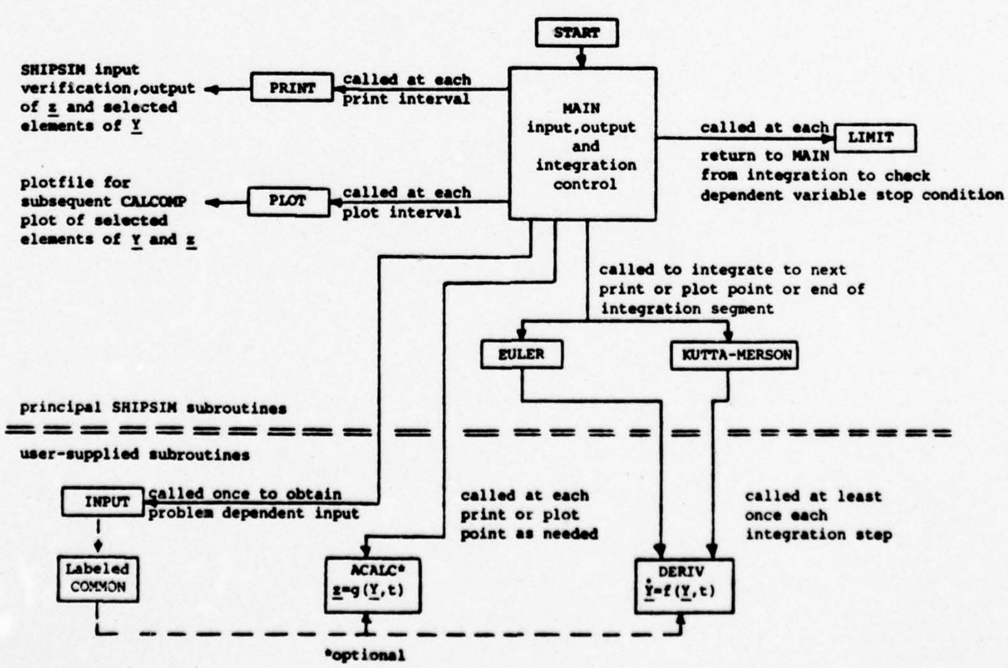


Figure 2. Macro Flow Chart for SHIPSIM

The SHIPSIM program has been written to be independent of the Michigan Terminal System (MTS) as much as possible. The principal exception is the plot option which uses many subroutines from the *PLOTSYS public file on MTS[16] to prepare the CALCOMP plotfile. Most computer installations have comparable software or the program can be used with the plot option removed. Appendix A contains the User's Documentation for SHIPSIM. Appendix D contains the Programmer's Documentation for SHIPSIM. Appendix F contains a source code listing for SHIPSIM. The plot option portions of the code which must be altered or removed to implement the program on another system without the plot capability are identified in Appendix D.

IV. Description of OPTSIM

OPTSIM is a group of subroutines which constitute the INPUT and DERIV subroutines needed by SHIPSIM. OPTSIM running under the control of SHIPSIM allows the simulation of the response of stationary, linear optimal control systems to specific user-specified process disturbances and/or initial condition errors while subject to measurement white noise. The input data set is constructed so that input used for system design using our version of the OPTSYS program (Appendix C) and the gain matrices as output by OPTSYS can be used directly with OPTSIM. The simulations can thus be obtained with a minimum of additional data preparation and programming by the user. The macro structure of OPTSIM running under the control of SHIPSIM is shown in Figure 3.

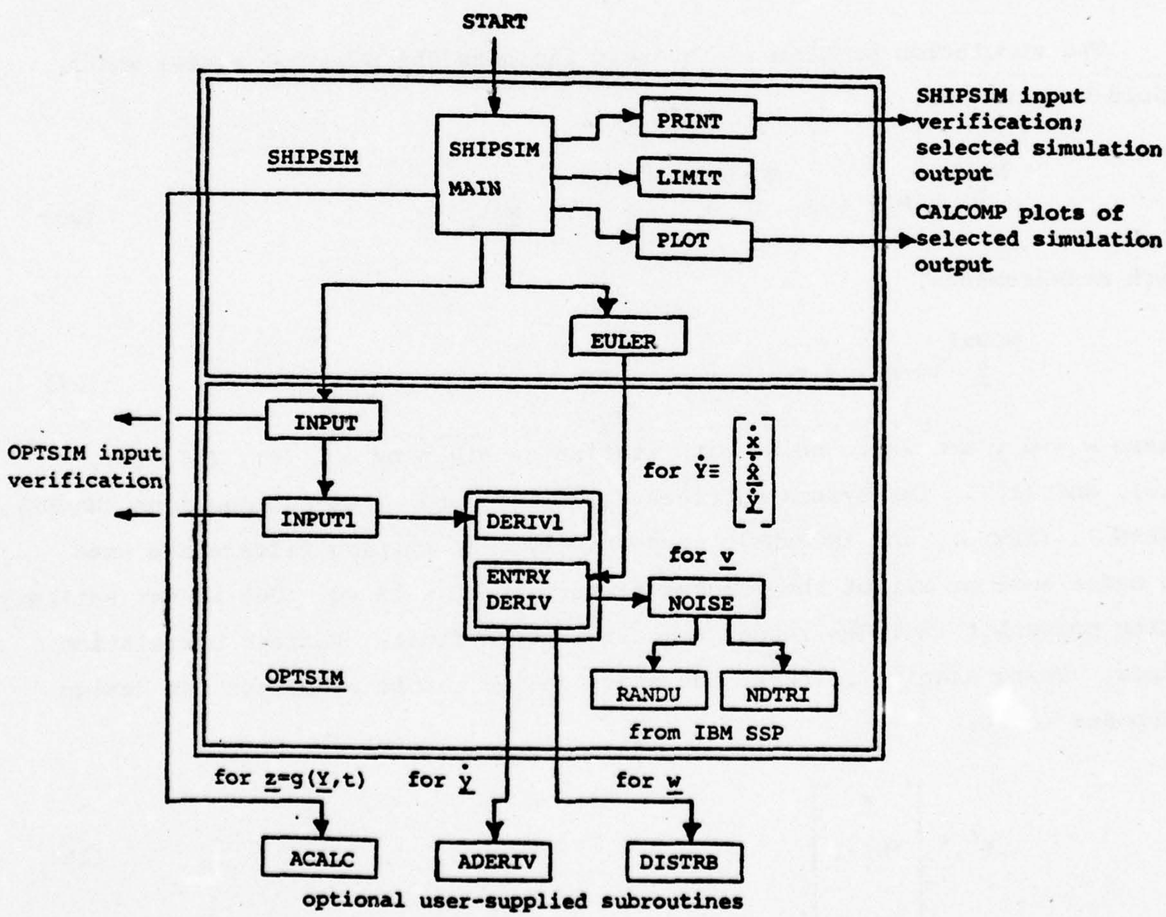


Figure 3. Macro Flow Chart for SHIPSIM/OPTSIM

A control system response to initial condition errors while subject to measurement noise can be simulated without any additional programming by the user. The initial conditions can be input as part of the SHIPSIM input. Measurement white noise $\underline{v}$ with user-specified standard deviation $\underline{\sigma}'$ is calculated within OPTSIM by subroutine NOISE which utilizes two subroutines from the IBM Scientific Subroutine Package[17] for randum number generation (RANDU) and transformation to normally-distributed quantities (NDTRI). If the user wishes to subject the system to a specific process disturbance $\underline{w}$, he must prepare a subroutine DISTRB which defines the process disturbance as a function of the states and/or time. The user can also include a subroutine ADERIV if quantities $\underline{y}$ in addition to the states $\underline{x}$ and estimates $\hat{\underline{x}}$ are to be integrated. Finally, the user can also include a subroutine ACALC (defined as part of SHIPSIM) if additional non-integrated quantities are desired at the print or plot points.

The simulation problem of interest includes the physical system which could be given by,

$$\overset{NSx1}{\dot{\underline{x}}} = F_S\underline{x} + G_S \overset{NCx1}{\underline{u}} + \Gamma \overset{NGx1}{\underline{w}} , \quad \underline{x}(t_o)=\underline{x}_o, \tag{26}$$

with measurements,

$$\overset{NOBx1}{\underline{z}} = H_S\underline{x} + \underline{v} , \tag{27}$$

where $\underline{w}$ and $\underline{v}$ are white noise with statistics given by eq. (4), (5), (9), (10), and (11). The system matrices F_S, G_S, Γ, and H_S have dimensions (NSxNS), (NSxNC), (NSxNG), and (NOBxNS), respectively. If shaping filters are used to model some or all of the process disturbances, $\underline{w}$ in eq. (26) is not entirely white noise but includes random quantities with finite, nonzero correlation times. Using shaping filters, the state vector can be augmented for design purposes to be,

$$\underline{x}' \equiv \begin{bmatrix} \underline{x} \\ x_{NS+1} \\ \vdots \\ x_{NE} \end{bmatrix} \tag{28}$$

and the design system equation or estimator design equation is given by,

$$\overset{NEx1}{\underline{\dot{x}}'} = F_e\underline{x}' + G_e\underline{u} + \Gamma'\underline{w}' \quad , \tag{29}$$

with measurements,

$$\underline{z} = H_e\underline{x}' + \underline{v} \quad , \tag{30}$$

where $\underline{w}'$ is now white noise with statistics given by eq. (4), (5), and (11). The estimator design matrices F_e, G_e, and H_e now have dimension (NExNE), (NExNC), and (NOBxNE), respectively.

The optimal state estimator design produces the Kalman-Bucy filter equation,

$$\dot{\hat{\underline{x}}} = F_e\hat{\underline{x}} + G_e\underline{u} + K(\underline{z}-H_e\hat{\underline{x}}) \quad , \tag{31}$$

where the filter gain matrix has dimension (NExNOB). The optimal controller design produces the control equation,

$$\underline{u} = C\hat{\underline{x}} \quad , \tag{32}$$

where the control gain matrix has dimension (NCxNE). Substituting eq. (27) and (32) into eq. (26) and (31) yields the system of equations to be simulated; i.e.,

$$\dot{\underline{x}} = F_s\underline{x} + G_sC\hat{\underline{x}} + \Gamma\underline{w} \quad , \tag{33}$$

$$\dot{\hat{\underline{x}}} = F_e\hat{\underline{x}} + G_eC\hat{\underline{x}} + KH_s\underline{x} + K\underline{v} - KH_e\hat{\underline{x}} \quad . \tag{34}$$

Rearranging these gives,

$$\begin{bmatrix} \underline{x} \\ \dot{\hat{\underline{x}}} \end{bmatrix} = \begin{bmatrix} F_s & G_sC \\ KH_s & F_e+G_eC-KH_e \end{bmatrix} \begin{bmatrix} \underline{x} \\ \hat{\underline{x}} \end{bmatrix} + \begin{bmatrix} \Gamma\underline{w} \\ K\underline{v} \end{bmatrix} = A \begin{bmatrix} \underline{x} \\ \hat{\underline{x}} \end{bmatrix} + \underline{b} \quad . \tag{35}$$

If no shaping filters are used, NS=NE and eq. (26) and (29) are identical so $F_s=F_e$, $G_s=G_e$, and $H_s=H_e$.

The simulation problem may also include integration of a number of additional quantities given by,

$$\overset{NA\times 1}{\dot{\underline{y}}} = \underline{h}(\underline{x}(t),\hat{\underline{x}}(t),\underline{y}(t),t) \quad . \tag{36}$$

SHIPSIM integrates the total system of NEQ=NS+NE+NA equations given by eq. (35) and (36) grouped as follows:

$$\overset{NEQ\times 1}{\dot{\underline{Y}}} \equiv \begin{bmatrix} \dot{\underline{x}} \\ \dot{\hat{\underline{x}}} \\ \dot{\underline{y}} \end{bmatrix} \quad . \tag{37}$$

The derivatives eq. (35) are calculated by OPTSIM using the user-supplied disturbance $\underline{w}$ from subroutine DISTRB. The derivatives eq. (36) are calculated in the user-supplied subroutine ADERIV. The combined vector of derivatives eq. (37) is loaded in OPTSIM and returned to the SHIPSIM integration subroutine.

The selection of integration method and step-size and the specification of white noise standard deviation $\underline{\sigma}'$ must be done carefully when simulating *stochastic systems. In design, the usual approach in modeling random* disturbances or noise which vary rapidly compared to the dominant time constants of the system is to first assume they are exponentially correlated. The correlation time τ_c and standard deviation σ are then obtained or approximated. For an exponentially correlated quantity the power spectral density is given by,[5]

$$Q = \frac{2\sigma^2\beta}{\omega^2+\beta^2} \quad , \tag{38}$$

where $\beta=\tau_c{}^{-1}$. This power spectral density is approximately constant for $\omega<<\beta$. The power spectral density can therefore be approximated by,

$$Q \cong 2\sigma^2/\beta = 2\sigma^2\tau_c \quad , \tag{39}$$

since τ_c is small compared to the dominant time constants of the system; i.e. $\omega<<\beta$. The noise is thus considered white noise with constant power spectral density given by eq. (39).

In simulating stochastic control systems, the continuous gauss-markov process eq. (35) is usually approximated by a discrete gauss-markov sequence

through the use of Euler or rectangular integration. In this approximation, care must be execised to ensure that the system covariances are preserved. The discrete gauss-markov sequence can be given by,

$$\underline{x}_{i+1} = \Phi_i \underline{x}_i + \Gamma_i \underline{w}_i \ , \qquad i = 0,1,\ldots,N \tag{40}$$

$$\underline{z}_i = H_i \underline{x}_i + \underline{v}_i \ , \qquad i = 0,1,\ldots,N \tag{41}$$

$$E[(\underline{w}_i - \bar{\underline{w}}_i)(\underline{w}_j - \bar{\underline{w}}_j)^T] = Q_i \delta_{ij} \ , \tag{42}$$

$$E[v_i v_j{}^T] = R_i \delta_{ij} \ , \tag{43}$$

$$E[(w_i - \bar{\underline{w}}_i) v_j{}^T] = E[(\underline{w}_i - \bar{\underline{w}}_i)(\underline{x}_o - \bar{\underline{x}}_o)^T] = E[v_i (\underline{x}_o - \bar{\underline{x}}_o)^T] = 0 \ , \tag{44}$$

where δ_{ij} is the Kronecker delta function. The control can be omitted in this discussion without loss of generality. If we take P_i as the error covariance of the discrete Kalman filter estimate at point i, the covariance can be propagated to the next measurement point i+1 using a time update to give the error covariance prior to the measurements M_{i+1} and then the measurements can be processed in the measurement update to give P_{i+1} . Governing update equations can be expressed as,

$$M_{i+1} = \Phi_i P_i \Phi_i^T + \Gamma_i Q_i \Gamma_i^T \quad \text{(time update)} \tag{45}$$

$$P_{i+1} = M_{i+1} - K_{i+1} H_{i+1} M_{i+1} \quad \text{(measurement update)} \tag{46}$$

where,

$$K_i = P_i H_i R_i^{-1} = M_i H_i^T (H_i M_i H_i^T + R_i)^{-1} \ . \tag{47}$$

(See Bryson and Ho, Applied Optimal Control,[3] pp. 349-351, 357, 360-361)

The corresponding equations for the continuous gauss-markov process and the error covariance are given by,

$$\dot{\underline{x}} = F\underline{x} + \Gamma\underline{w} \ , \qquad \underline{x}(t_0) = \underline{x}_0 \ , \tag{48}$$

$$\underline{z} = H\underline{x} + \underline{v} \ , \tag{49}$$

$$E[(\underline{w}(t) - \bar{\underline{w}}(t))(\underline{w}(\tau) - \bar{\underline{w}}(\tau))^T] = Q\delta(t-\tau) \ , \tag{50}$$

$$E[\underline{v}(t)\underline{v}(\tau)^T] = R\delta(t-\tau) \ , \tag{51}$$

$$\dot{P} = FP + PF^T + \Gamma Q \Gamma^T - KHP \quad , \tag{52}$$

where eq. (18) has been used in eq. (19) and eq. (4), (6), (9), and (11) also apply. When Euler integration with integration step-size Δt is utilized, $x(t+\Delta t)$ is approximated using eq. (48) to be,

$$\underline{x}(t+\Delta t) = \underline{x}(t) + F\underline{x}(t)\Delta t + \Gamma \underline{w}(t)\Delta t. \tag{53}$$

By comparison between eq. (40) and eq. (53) we must then have,

$$\Phi_i = [I + F\Delta t] \quad , \tag{54}$$

$$\Gamma_i = \Gamma \Delta t \quad . \tag{55}$$

Again when using Euler integration, the error covariance $P(t+\Delta t)$ is in effect approximated using eq. (52) to be,

$$P(t+\Delta t) = P(t) + FP(t)\Delta t + P(t)F^T\Delta t + \Gamma Q \Gamma^T \Delta t - K(t)HP(t)\Delta t. \tag{56}$$

This should produce the same error covariance as eq. (45) and (46) for the simulation to be correct.

Using eq. (54) and (55) in eq. (45) yields,

$$M_{i+1} = P_i + FP_i\Delta t + P_iF^T\Delta t + FP_iF^T\Delta t^2 + \Gamma(Q_i\Delta t)\Gamma^T\Delta t, \tag{57}$$

where the term in Δt^2 can be neglected since the integration step-size Δt is kept small. Using eq. (57) in eq. (46) now yields,

$$P_{i+1} = P_i + FP_i\Delta t + P_iF^T\Delta t + \Gamma(Q_i\Delta t)\Gamma^T\Delta t - \frac{K_{i+1}}{\Delta t} H(P_i + \text{higher order terms in } \Delta t)\Delta t. \tag{58}$$

In the limit as $\Delta t \to 0$, $K_{i+1} \to K_i$ so we have,

$$\frac{K_{i+1}}{\Delta t} = \frac{K_i}{\Delta t} = P_iH^T(R_i\Delta t)^{-1} \quad , \tag{59}$$

and eq. (56) will equal eq. (58) if we take,

$$Q_i\Delta t = Q, \tag{60}$$

$$R_i\Delta t = R \ . \tag{61}$$

The correct measurement noise covariance R_i to use in the simulation therefore depends on the integration step-size Δt. If σ_i' is the standard deviation for noise element i (square root of element i of R_i), a correct simulation must utilize,

$$\sigma_i' = \left[\frac{R_{ii}}{\Delta t}\right]^{1/2} , \qquad (62)$$

where R_{ii} is the i-th diagonal element of R used in the design of the continuous stochastic control system and Δt is the fixed integration step-size used with Euler integration. Notice that the standard deviation used in eq. (39) to approximate the power spectral density for the measurement noise in the design of the continuous system; i.e.,

$$R_{ii} \cong 2\sigma_i^{\,2}\tau_c , \qquad (63)$$

is not the same as the standard deviation σ_i' which must be used in the simulation using Euler integration.

User's Documentation for OPTSIM is included in Appendix B; Programmer's Documentation for OPTSIM is included in Appendix E; a source code listing of the OPTSIM subroutines is included in Appendix G. An example simulation of the response of an optimal stochastic path controller for a tanker using SHIPSIM/OPTSIM is described in the next section.

V. Tanker Path Control Example

This section will briefly present the simulation of an optimal stochastic path controller for a tanker operating in shallow water as an example of the use of the SHIPSIM/OPTSIM simulation program. For a complete development and extensive results concerning this problem the reader should consult our research report on the optimal stochastic path control of surface ships in shallow water.[1] Considering the small-deviation control of a ship along a path using the coordinate system shown in Figure 4, the equations of motion in the horizontal plane can be written as follows in non-dimensional form:

$$\frac{d\psi'}{dt'} = r', \tag{64}$$

$$(I'_{zz}+J'_{zz})\frac{dr'}{dt'} = N_{\beta'}\beta' + N_{r'}r' + N_{\dot{\beta}'}\dot{\beta}' + N_{\delta'}\delta' \,, \tag{65}$$

$$-(m'+m_y')\frac{d\beta'}{dt'} = Y_{\beta'}\beta' + (-m'+Y_r')r' + Y_{\dot{r}'}\dot{r}' + Y_{\delta'}\delta' \,, \tag{66}$$

$$\frac{d\eta'}{dt'} = \psi' - \beta' \,. \tag{67}$$

Numerical values for the coefficients in eq. (65) and (66) for the 150,000 DWT tanker *Tokyo Maru* at 12 knots were given by Fujino[18] for various values of water depth to draft ratio (H/T). Equations (65) and (66) can be solved simultaneously[1] to produce two equivalent equations without the coupling in $\dot{r}'$ and $\dot{\beta}'$. Finally the rudder control system can be modeled by a first-order system,

$$\frac{d\delta'}{dt'} = \frac{1}{T_r}(\delta_c'-\delta') \,, \tag{68}$$

where δ_c' is the commanded rudder angle. These equations yield the system equations,

$$\frac{d}{dt'}\begin{bmatrix}\psi'\\ r'\\ \beta'\\ \eta'\\ \delta'\end{bmatrix} = \begin{bmatrix}0 & 1 & 0 & 0 & 0\\ 0 & f_{22} & f_{23} & 0 & f_{25}\\ 0 & f_{32} & f_{33} & 0 & f_{35}\\ 1 & 0 & -1 & 0 & 0\\ 0 & 0 & 0 & 0 & -1/T_r\end{bmatrix}\begin{bmatrix}\psi'\\ r'\\ \beta'\\ \eta'\\ \delta'\end{bmatrix} + \begin{bmatrix}0\\ 0\\ 0\\ 0\\ 1/T_r\end{bmatrix}\delta_c' \,. \tag{69}$$

We also have the equation for the movement along the path which when non-dimensionalized becomes just,

$$\frac{d\xi'}{dt'} = 1 \quad . \tag{70}$$

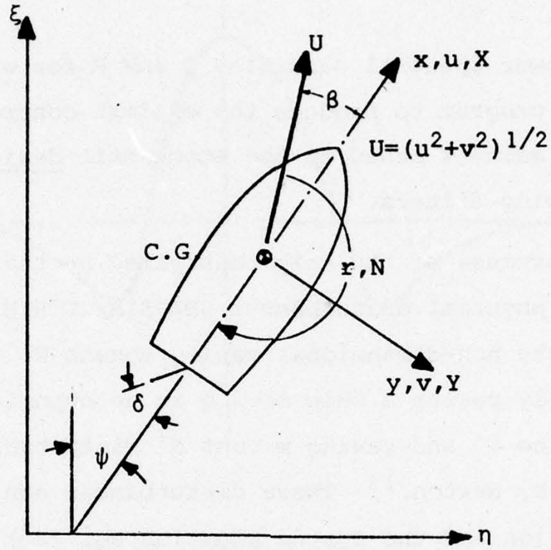


Figure 4. Ship Path Control Coordinate System

The effects of disturbances such as passing ships, bank effects, spatial current changes, etc. can be modeled as exponentially correlated disturbances with the use of first-order shaping filters,

$$T_N \frac{dN'}{dt'} + N' = w_N \quad , \tag{71}$$

$$T_Y \frac{dY'}{dt'} + Y' = w_Y \quad , \tag{72}$$

where N' is a yawing moment and Y' is a lateral force acting on the ship. Augmenting the state vector, the estimator design equation eq. (29) becomes,

$$\frac{d}{dt'}\begin{bmatrix}\psi'\\ r'\\ \beta'\\ \eta'\\ \delta'\\ N'\\ Y'\end{bmatrix} = \begin{bmatrix}0 & 1 & 0 & 0 & 0 & 0 & 0\\ 0 & f_{22} & f_{23} & 0 & f_{25} & f_{26} & f_{27}\\ 0 & f_{32} & f_{33} & 0 & f_{35} & f_{36} & f_{37}\\ 1 & 0 & -1 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & -1/T_r & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & -1/T_N & 0\\ 0 & 0 & 0 & 0 & 0 & 0 & -1/T_Y\end{bmatrix}\begin{bmatrix}\psi'\\ r'\\ \beta'\\ \eta'\\ \delta'\\ N'\\ Y'\end{bmatrix} + \begin{bmatrix}0\\ 0\\ 0\\ 0\\ 1/T_r\\ 0\\ 0\end{bmatrix}\delta'_c + \begin{bmatrix}0 & 0\\ 0 & 0\\ 0 & 0\\ 0 & 0\\ 0 & 0\\ 1/T_N & 0\\ 0 & 1/T_Y\end{bmatrix}\begin{bmatrix}w_N\\ w_Y\end{bmatrix} , \tag{73}$$

and the measurement could be ψ' from a compass, r' from a rate gyro and η' from radar (or DECCA) to give the measurement equation eq. (30),

$$\underline{z} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \underline{x}' + \underline{v} \, . \tag{74}$$

These equations with power spectral densities Q and R for $\underline{w}$ and $\underline{v}$ were used as input to the OPTSYS program to produce the optimal control gains C and the optimal state observer gains K based on the stochastic design disturbances represented by the shaping filters.

To simulate the response of the ship controlled by the optimal stochastic controller to specific physical disturbances SHIPSIM/OPTSIM can be used. Figure 5 shows, for example, the non-dimensional yawing moment N' which would act on the ship as it closely passes a ship moving in an opposite, parallel direction. Lateral force Y' and yawing moment N' disturbances were developed based on data obtained by Newton.[19] These disturbances can be applied to the ship in the simulation and the system equation eq. (26) then becomes,

$$\frac{d}{dt'} \begin{bmatrix} \psi' \\ r' \\ \beta' \\ \eta' \\ \delta' \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & f_{22} & f_{23} & 0 & f_{25} \\ 0 & f_{32} & f_{33} & 0 & f_{35} \\ 1 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1/T_r \end{bmatrix} \begin{bmatrix} \psi' \\ r' \\ \beta' \\ \eta' \\ \delta' \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1/T_r \end{bmatrix} \delta_c' + \begin{bmatrix} 0 & 0 \\ f_{26} & f_{27} \\ f_{36} & f_{37} \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} N' \\ Y' \end{bmatrix} , \tag{75}$$

with the measurement equation eq. (27),

$$\underline{z} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} \underline{x} + \underline{v} \, . \tag{76}$$

Eq. (73), (74), (75), and (76) define the system and estimator design matrices F_s, G_s, H_s, F_e, G_e, H_e, and Γ. The OPTSYS output defines the gain matrices C and K. The measurement noise simulation standard deviations $\underline{\sigma}'$ were derived from R using eq. (62). The simulation equation eq. (35) is thus fully defined. Equation (70) must be treated as an additional derivative to be integrated along with eq. (35).

To set up the SHIPSIM/OPTSIM simulation subroutines DISTRB and ADERIV must be prepared. The disturbances as in Fig. 5 were curve fit with piece-

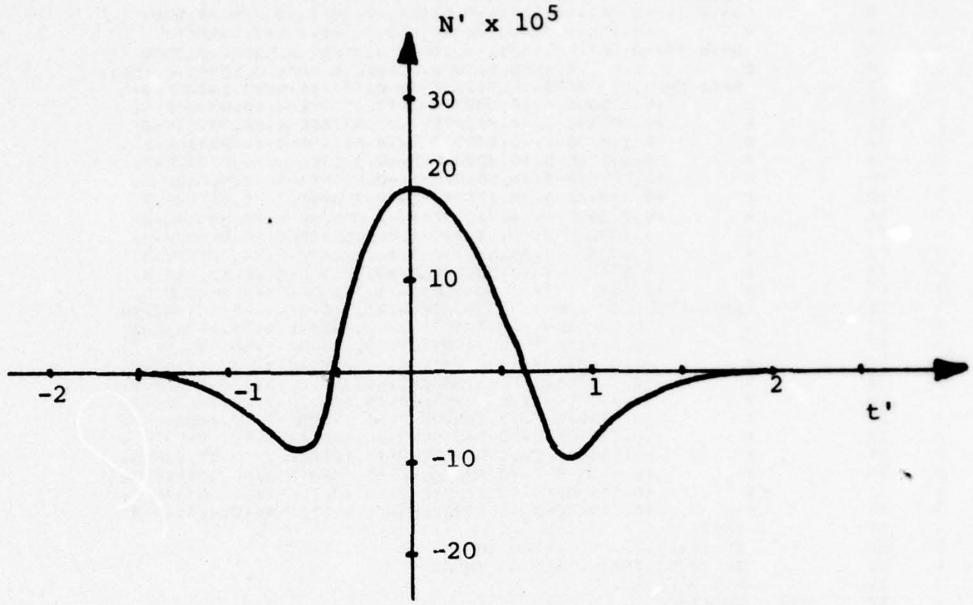


Figure 5. Yawing Moment N' due to Passing Ship

wise cubic splines. The subroutine DISTRB can then take the current value of non-dimensional t=TIME, locate which segment K of each spline is applicable by testing for $t'_K \leq TIME < t'_{K+1}$, and then evaluate the splines to produce N'(TIME) and Y'(TIME). These disturbances can then be returned in $\underline{w}$. A listing for the DISTRB subroutine for a passing ship disturbance is shown in Fig. 6. The subroutine ADERIV for NA=1 to implement eq. (70) requires only one line of executable code YDOT(1)=1. A listing will not be included here. There is no need to prepare the optional subroutine ACALC for this simulation.

The SHIPSIM input data set for a simulation of the tanker beginning with zero initial conditions and then sailing past a ship at t'=7. is shown in Fig. 7. This data should be reviewed in conjunction with reference to Appendix A User's Documentation for SHIPSIM. Three integration segments are utilized to obtain more closely spaced printed output over the interval $4.0 \leq t' \leq 9.0$ where the disturbance occurs. The Euler integration step-size is 0.005 t' units (time to travel one ship length, t'=t/U). Printed output of ψ', r', β', η', δ', $\hat{\psi}'$, $\hat{N}'$, $\hat{Y}'$, and X' are requested. Plots of η', $\hat{N}'$,

```
      SUBROUTINE DISTRB(T,X,W,NG,NFQ)
      IMPLICIT REAL*8 (A-H,O-$)
      DIMENSION W(2),X(25)
      DIMENSION TN(13),CN(4,12),TY(13),CY(4,12)
      DATA TN/-0.2E1,-0.15E1,-0.125E1,-0.10E1,-0.9E0,-0.5E0,
     *        -0.1E0,0.2E0,0.4E0,0.5E0,0.6E0,0.1E1,0.15E1/
      DATA TY/-0.2E1,-0.15E1,-0.10E1,-0.75E0,-0.50E0,-0.25E0,
     *         0.0E0,0.25E0,0.50E0,0.70E0,0.90E0,0.125E1,0.15E1/
      DATA CN/0.29370E-4,+0.21261E-4,-0.73424E-5,+0.14685E-4,
     *       +0.42522E-4,+0.58739E-3,+0.37342E-4,+0.83288E-4,
     *       +0.58739E-3,-0.47209E-3,+0.83288E-4,+0.34951E-3,
     *       -0.11802E-2,-0.54097E-2,+0.81180E-3,+0.95410E-3,
     *       -0.13524E-2,+0.32983E-3,+0.44139E-3,-0.22777E-3,
     *       +0.32983E-3,+0.50187E-3,-0.22777E-3,-0.58030E-3,
     *       +0.66916E-3,+0.17539E-2,-0.72689E-3,-0.65785E-3,
     *       +0.26308E-2,-0.76323E-4,-0.85523E-3,+0.30529E-5,
     *       -0.15265E-3,-0.14607E-1,+0.15265E-5,+0.84607E-3,
     *       -0.14607E-1,-0.14179E-2,+0.84607E-3,+0.81418E-3,
     *       -0.35447E-3,+0.23664E-3,+0.25672E-3,+0.12138E-4,
     *       +0.18931E-3,-0.14655E-4,-0.73275E-5,+0.36638E-5/
      DATA CY/-0.66367E-4,-0.27267E-4,+0.16592E-4,-0.33183E-4,
     *        -0.27267E-4,-0.30457E-3,-0.33183E-4,-0.12386E-3,
     *        -0.60913E-3,+0.24839E-2,-0.36193E-3,-0.79524E-3,
     *        +0.24839E-2,+0.47536E-2,-0.79524E-3,-0.29710E-3,
     *        +0.47536E-2,-0.61385E-2,-0.29710E-3,+0.19837E-2,
     *        -0.61385E-2,-0.67983E-3,+0.19837E-2,+0.19625E-2,
     *        -0.67983E-3,-0.13822E-2,+0.19625E-2,+0.16864E-2,
     *        -0.13822E-2,-0.14712E-2,+0.16864E-2,+0.89195E-3,
     *        -0.18390E-2,+0.54353E-2,+0.10736E-2,-0.21741E-3,
     *        +0.54353E-2,+0.97685E-4,-0.21741E-3,-0.20391E-3,
     *        +0.55820E-4,+0.14889E-3,-0.12112E-3,-0.75382E-4,
     *        +0.20844E-3,-0.74422E-3,-0.93028E-4,+0.46514E-4/
      TP=7.
      IF (T.LT.TP+TN(1)) GO TO 30
      IF (T.GT.TP+TN(13)) GO TO 30
      K=1
   10 CONTINUE
      IF (T.GE.TP+TN(K).AND.T.LE.TP+TN(K+1)) GO TO 20
      K=K+1
      GO TO 10
   20 CONTINUE
      WN=CN(1,K)*(TP+TN(K+1)-T)**3+CN(2,K)*(T-TP-TN(K))**3
     *  +CN(3,K)*(TP+TN(K+1)-T)+CN(4,K)*(T-TP-TN(K))
      GO TO 40
   30 CONTINUE
      WN=0.
   40 CONTINUE
      IF (T.LT.TP+TY(1)) GO TO 70
      IF (T.GT.TP+TY(13)) GO TO 70
      K=1
   50 CONTINUE
      IF (T.GE.TP+TY(K).AND.T.LE.TP+TY(K+1)) GO TO 60
      K=K+1
      GO TO 50
   60 CONTINUE
      WY=CY(1,K)*(TP+TY(K+1)-T)**3+CY(2,K)*(T-TP-TY(K))**3
     *  +CY(3,K)*(TP+TY(K+1)-T)+CY(4,K)*(T-TP-TY(K))
      GO TO 80
   70 CONTINUE
      WY=0.
   80 CONTINUE
      W(1)=WN
      W(2)=WY
      RETURN
      END
#END OF FILE
#
```

Figure 6. Subroutine DISTRB for Passing Ship Disturbance

```
    3    0    1    3   1.0
T-M,NOISE,PASSING SHIP,H/T=1.89,OPTIMAL,SHAPING FILTER
PSI        PSIDOT     BETA       ETA        DELTA      PSI$
ETA$       DELTA$     N$         Y$         X
0.0        0.0        0.0        0.0        0.0        0.0
0.0        0.0        0.0        0.0        0.0        0.0
NONE
PSI      PSIDOT  BETA      ETA       DELTA    PSI$     N$        Y$
ETA        DELTA       N$
    1        4.0    .0050      0.5  0.1
    1        9.0    .0050      0.1  0.1
    1       30.0    .0050      0.5  0.1
```

Figure 7. SHIPSIM Input Data Set for Example Simulation

and δ' are requested using a scale factor SF=1.0 to produce 8-1/2"x11" CALCOMP plots. The OPTSIM input data set for the simulation is shown in Fig. 8. This data should be reviewed in conjunction with reference to Appendix B User's Documentation for OPTSIM.

```
  1
  5  1  3  2  1  7
       0.0E0        1.0E0        0.0E0        0.0E0        0.0E0
 0.0        -0.17657E+01 0.57359E+01 0.0             -0.88074E+00
 0.0         0.17199E+00-0.52766E+00 0.0             -0.15607E+00
       1.0E0        0.0E0       -1.0E0        0.0E0        0.0E0
       0.0E0        0.0E0        0.0E0        0.0E0     -0.439E0
       0.0E0        1.0E0        0.0E0        0.0E0        0.0E0
       0.0E0
 0.0        -0.17657E+01 0.57359E+01 0.0             -0.88074E+00
-0.50043E+01
 0.0         0.17199E+00-0.52766E+00 0.0             -0.15607E+00
-0.28233E+02
       1.0E0        0.0E0       -1.0E0        0.0E0        0.0E0
       0.0E0
       0.0E0        0.0E0        0.0E0        0.0E0     -0.439E0
       0.0E0
       0.0E0        0.0E0        0.0E0        0.0E0        0.0E0
       0.0E0
       0.0E0        0.0E0        0.0E0        0.0E0        0.0E0
      -1.0E0
       0.0E0        0.0E0        0.0E0        0.0E0      0.439E0
       0.0E0        0.0E0        0.0E0        0.0E0      0.439E0
       0.0E0
 7.74621E+00 4.62370E+00 1.70009E+01 2.42523E+00-4.71320E+00
-2.25797E+02
 0.0         0.0         0.47768E+03-0.50043E+01 0.21141E+02
 0.0         0.0         0.0         0.0
       1.0E0        0.0E0        0.0E0        0.0E0        0.0E0
       1.0E0        0.0E0        0.0E0        0.0E0        0.0E0
       0.0E0        1.0E0        0.0E0
       1.0E0        0.0E0        0.0E0        0.0E0        0.0E0
       0.0E0        0.0E0        1.0E0        0.0E0        0.0E0
       0.0E0        0.0E0        0.0E0        0.0E0       0.0E0
       0.0E0        0.0E0        0.0E0
 2.18863E-01 1.00536E+00 1.42610E-03 4.83587E-02 3.29596E+01
 7.54479E-02 1.08091E+01-3.42418E-01 1.39170E-01-5.15903E+00
 0.0         0.0         0.0        -6.94613E-04 1.12292E+00
-6.31352E-04-5.98504E-01 6.63869E-03
   1.0541E-2    2.3116E-3    1.0413E-1
```

Figure 8. OPTSIM Input Data Set for Example Simulation

The printed output from the simulation is shown in Fig. 9. The output consists of the title printed by SHIPSIM, input verification of the OPTSIM input data produced by subroutines INPUT and INPUT1 in OPTSIM, input verification of the SHIPSIM input data, and then the actual simulation results. The plot output from the simulation is shown in Figures 10, 11, and 12. These plots are reduced here in size from the 8-1/2"x11" plots actually produced by the CALCOMP plotter. Figure 10 shows effective control system response to return the ship to the desired path (η'=0) following the disturbance

caused by the ship passing at t'=7. The maximum lateral offset is about 19 feet full scale. Figure 11 shows the estimate of the yawing moment disturbance $\hat{N}'$ produced by the Kalman-Bucy filter. This can be compared with the actual disturbance produced by DISTRB as shown in Figure 5. The validity of using first-order shaping filters to produce <u>design</u> disturbances which model this type <u>actual</u> disturbance is confirmed by a detailed comparison. Figure 12 shows the rudder usage during the simulation. Maximum value is about 4°. The cost of this simulation run was $3.07 with an additional $1.86 for the three plots.

Figure 9. SHIPSIM/OPTSIM Printed Output for Example Simulation

```
#RUN SGWY:SHIPSIM.O+OPTSIM.O+ADERIV.O+DISTRBP.O+ACALC.D+NAAS:SSP+*PLOTSYS 4=DATA41 5=DATA51 9=-PLOTFILE
#EXECUTION BEGINS


UNIVERSITY OF MICHIGAN DEPARTMENT OF NAVAL ARCHITECTURE AND MARINE ENGINEERING

SHIPSIM CONTINUOUS SYSTEMS SIMULATION PROGRAM


OPTSIM OPTIMAL STOCHASTIC CONTROLLER SIMULATION PROGRAM

INPUT VERIFICATION   NEQ = 13

ORDER OF SYSTEM =  5

NUMBER OF CONTROLS =  1

NUMBER OF OBSERVATIONS =  3

NUMBER OF PROCESS NOISE SOURCES =  2

NUMBER OF AUXILIARY STATES =  1

ORDER OF ESTIMATOR =  7

SYSTEM OPEN LOOP DYNAMICS MATRIX....FS(NS,NS)

  0.0           0.10000E+01   0.0           0.0           0.0
  0.0          -0.17657E+01   0.57359E+01   0.0          -0.88074E+00
  0.0           0.17199E+00  -0.52766E+00   0.0          -0.15607E+00
  0.10000E+01   0.0          -0.10000E+01   0.0           0.0
  0.0           0.0           0.0           0.0          -0.43900E+00

ESTIMATOR OPEN LOOP DYNAMICS MATRIX....FE(NE,NE)

  0.0           0.10000E+01   0.0           0.0           0.0           0.0           0.0
  0.0          -0.17657E+01   0.57359E+01   0.0          -0.88074E+00   0.47768E+03  -0.50043E+01
  0.0           0.17199E+00  -0.52766E+00   0.0          -0.15607E+00   0.21141E+02  -0.28233E+02
  0.10000E+01   0.0          -0.10000E+01   0.0           0.0           0.0           0.0
  0.0           0.0           0.0           0.0          -0.43900E+00   0.0           0.0
  0.0           0.0           0.0           0.0           0.0          -0.10000E+01   0.0
  0.0           0.0           0.0           0.0           0.0           0.0          -0.10000E+01

SYSTEM CONTROL DISTRIBUTION MATRIX....GS(NS,NC)

  0.0
  0.0
  0.0
  0.0
  0.43900E+00
```

Figure 9. SHIPSIM/OPTSIM Printed Output for Example Simulation...Continued

```
ESTIMATOR CONTROL DISTRIBUTION MATRIX....GE(NE,NC)

   0.0
   0.0
   0.0
   0.0
   0.43900E+00
   0.0
   0.0

FEEDBACK CONTROL GAINS....C(NC,NE)

   0.77462E+01   0.46237E+01   0.17001E+02   0.24252E+01  -0.47132E+01   0.14691E+04  -0.22580E+03

SYSTEM DISTURBANCE DISTRIBUTION MATRIX....GAMMA(NS,NG)

   0.0           0.0
   0.47768E+03  -0.50043E+01
   0.21141E+02  -0.28233E+02
   0.0           0.0
   0.0           0.0

SYSTEM MEASUREMENT SCALING MATRIX....HS(NOB,NS)

   0.10000E+01   0.0           0.0           0.0           0.0
   0.0           0.10000E+01   0.0           0.0           0.0
   0.0           0.0           0.0           0.10000E+01   0.0

ESTIMATOR MEASUREMENT SCALING MATRIX....HE(NOB,NE)

   0.10000E+01   0.0           0.0           0.0           0.0           0.0           0.0
   0.0           0.10000F+01   0.0           0.0           0.0           0.0           0.0
   0.0           0.0           0.0           0.10000E+01   0.0           0.0           0.0

KALMAN-BUCY FILTER GAINS....K(NE,NOB)

   0.21886E+00   0.10054E+01   0.14261E-02
   0.48359E-01   0.32960E+02  -0.25429E-02
   0.75448E-01   0.10809E+02  -0.34242E+00
   0.13917E+00  -0.51590E+01   0.82120E+00
   0.0           0.0           0.0
  -0.69461E-03   0.11229E+01   0.40023E-02
  -0.63135E-03  -0.59850E+00   0.66387E-02

MEASUREMENT NOISE STANDARD DEVIATIONS....SIGMA(NOB)

   0.10541E-01
   0.23116E-02
   0.10413E+00


T-M,NOISE,PASSING SHIP,H/T=1.89,OPTIMAL,SHAPING FILTER
```

Figure 9. SHIPSIM/OPTSIM Printed Output for Example Simulation...Continued

* * * VARIABLES AND INITIAL VALUES:

```
PSI     = 0.0     PSIDOT   = 0.0     BETA     = 0.0     ETA      = 0.0
DELTA   = 0.0     PSI$     = 0.0     PSIDOT$  = 0.0     BETA$    = 0.0
ETA$    = 0.0     DELTA$   = 0.0     N$       = 0.0     Y$       = 0.0
X       = 0.0
```

* * * VARIABLES TO BE PRINTED: PSI PSIDOT BETA ETA DELTA PSI$ N$ Y$ X

* * * VARIABLES TO BE PLOTTED: ETA DELTA N$

* * * INTEGRATION CONTROL PARAMETERS:

SEGMENT	METHOD	TF	PRD	PLD	FIRSTP	EPS	AB	NCUTS
1	EULER	0.4000E+01	0.5000E+00	0.1000E+00	0.5000E-02			
2	EULER	0.9000E+01	0.1000E+00	0.1000E+00	0.5000E-02			
3	EULER	0.3000E+02	0.5000E+00	0.1000E+00	0.5000E-02			

* * * *INTEGRATION SEGMENT 1*

TIME	PSI	PSIDOT	BETA	ETA	DELTA	PSI$	N$	Y$	X
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.5000E+00	-.9059E-05	0.3096E-04	0.4382E-05	-.9949E-06	-.2449E-03	-.1407E-03	-.1230E-04	0.7048E-05	0.5000E+00
0.1000E+01	0.4418E-04	0.2428E-03	0.5151E-04	-.7585E-05	0.1033E-02	-.2621E-03	-.2933E-05	0.2651E-05	0.1000E+01
0.1500E+01	0.1780E-03	0.3336E-03	0.8376E-04	0.1168E-04	-.4549E-03	-.1245E-03	0.1271E-04	-.6041E-05	0.1500E+01
0.2000E+01	0.2996E-03	0.2010E-03	0.6391E-04	0.9949E-04	-.6329E-03	-.4574E-04	-.2272E-04	0.1328E-04	0.2000E+01
0.2500E+01	0.3724E-03	-.8333E-04	0.7526E-05	0.2469E-03	0.4878E-03	0.1082E-03	-.1479E-04	0.9908E-05	0.2500E+01
0.3000E+01	0.3481E-03	0.1840E-03	0.4060E-04	0.4200E-03	-.1701E-02	-.1475E-03	0.1817E-04	-.7716E-05	0.3000E+01
0.3500E+01	0.4920E-03	0.2388E-03	0.6721E-04	0.5953E-03	0.1052E-02	0.4344E-04	0.5294E-05	-.8908E-06	0.3500E+01
0.4000E+01	0.5649E-03	0.6889E-04	0.3382E-04	0.8371E-03	0.2869E-03	0.1125E-03	-.8173E-05	0.6329E-05	0.4000E+01

* * * INTEGRATION SEGMENT 2

TIME	PSI	PSIDOT	BETA	ETA	DELTA	PSI$	N$	Y$	X
0.4000E+01	0.5649E-03	0.6889E-04	0.3382E-04	0.8371E-03	0.2869E-03	0.1125E-03	-.8173E-05	0.6329E-05	0.4000E+01
0.4100E+01	0.5708E-03	0.3550E-04	0.2596E-04	0.8907E-03	0.8919E-03	0.3332E-04	0.1665E-04	-.6225E-05	0.4100E+01
0.4200E+01	0.5707E-03	-.4699E-04	0.8479E-05	0.9461E-03	0.1473E-02	0.1495E-03	0.3854E-05	0.3833E-06	0.4200E+01
0.4300E+01	0.5604E-03	-.1729E-03	-.1830E-04	0.1003E-02	0.2014E-02	0.2063E-03	0.6913E-05	-.1010E-05	0.4300E+01
0.4400E+01	0.5366E-03	-.2957E-03	-.4655E-04	0.1061E-02	0.9563E-03	0.2283E-03	-.2733E-04	0.1708E-04	0.4400E+01
0.4500E+01	0.5060E-03	-.3015E-03	-.5443E-04	0.1119E-02	-.1861E-03	0.2229E-03	0.1559E-04	-.5956E-05	0.4500E+01
0.4600E+01	0.4763E-03	-.3074E-03	-.6117E-04	0.1174E-02	0.5830E-03	0.1569E-03	-.2614E-05	0.4012E-05	0.4600E+01
0.4700E+01	0.4445E-03	-.3309E-03	-.7063E-04	0.1226E-02	0.5282E-03	0.3874E-04	-.1184E-04	0.9358E-05	0.4700E+01
0.4800E+01	0.4108E-03	-.3357E-03	-.7637E-04	0.1277E-02	0.3698E-04	-.4255E-04	-.1879E-04	0.1319E-04	0.4800E+01
0.4900E+01	0.3773E-03	-.3643E-03	-.8561E-04	0.1324E-02	0.1768E-02	0.3879E-04	0.2250E-04	-.8661E-05	0.4900E+01
0.5000E+01	0.3343E-03	-.4750E-03	-.1107E-03	0.1370E-02	0.3443E-03	-.1323E-04	-.3671E-04	0.2283E-04	0.5000E+01
0.5100E+01	0.2881E-03	-.4426E-03	-.1102E-03	0.1412E-02	0.1171E-03	0.5135E-04	0.6059E-05	-.3729E-06	0.5100E+01
0.5200E+01	0.2453E-03	-.4084E-03	-.1084E-03	0.1450E-02	0.5683E-03	-.1032E-04	0.1424E-05	0.2341E-05	0.5200E+01

Figure 9. SHIPSIM/OPTSIM Printed Output for Example Simulation...Continued

0.5300E+01	0.2080E-03	-.2980E-03	-.2218E-04	0.1483E-02	-.5551E-03	-.6271E-04	-.2847E-04	0.1802E-04	0.5300E+01
0.5400E+01	0.1909E-03	-.3310E-04	-.4597E-04	0.1509E-02	0.6126E-03	-.8302E-04	0.3202E-04	-.1392E-04	0.5400E+01
0.5500E+01	0.2026E-03	0.3310E-03	0.2056E-04	0.1531E-02	-.4024E-03	-.1857E-03	0.1018E-04	-.2292E-05	0.5500E+01
0.5600E+01	0.2591E-03	0.8272E-03	0.1139E-03	0.1547E-02	0.1866E-02	-.1443E-03	0.6483E-04	-.3096E-04	0.5600E+01
0.5700E+01	0.3637E-03	0.1321E-02	0.2006E-03	0.1562E-02	0.3858E-02	0.1304E-03	0.1005E-05	0.2769E-05	0.5700E+01
0.5800E+01	0.5366E-03	0.2296E-02	0.3474E-03	0.1579E-02	0.3767E-02	0.3543E-03	-.5008E-05	0.5611E-05	0.5800E+01
0.5900E+01	0.8400E-03	0.3952E-02	0.5777E-03	0.1601E-02	0.6757E-02	0.6909E-03	0.5690E-04	-.2719E-04	0.5900E+01
0.6000E+01	0.1334E-02	0.6097E-02	0.8763E-03	0.1635E-02	0.1125E-01	0.1299E-02	0.8419E-04	-.4167E-04	0.6000E+01
0.6100E+01	0.2055E-02	0.8456E-02	0.1251E-02	0.1696E-02	0.1466E-01	0.2049E-02	0.5273E-04	-.2510E-04	0.6100E+01
0.6200E+01	0.2997E-02	0.1028E-01	0.1671E-02	0.1800E-02	0.1865E-01	0.2921E-02	0.8716E-04	-.4341E-04	0.6200E+01
0.6300E+01	0.4049E-02	0.1040E-01	0.1993E-02	0.1965E-02	0.2336E-01	0.3895E-02	0.5419E-04	-.2571E-04	0.6300E+01
0.6400E+01	0.5009E-02	0.8327E-02	0.2074E-02	0.2212E-02	0.2488E-01	0.4791E-02	0.3190E-04	-.1381E-04	0.6400E+01
0.6500E+01	0.5659E-02	0.4144E-02	0.1778E-02	0.2550E-02	0.2372E-01	0.5613E-02	0.2687E-05	0.8294E-06	0.6500E+01
0.6600E+01	0.5807E-02	-.1684E-02	0.9810E-03	0.2983E-02	0.1823E-01	0.5764E-02	-.5189E-04	0.2947E-04	0.6600E+01
0.6700E+01	0.5322E-02	-.8456E-02	-.3274E-03	0.3507E-02	0.9038E-02	0.5231E-02	-.5083E-04	0.2848E-04	0.6700E+01
0.6800E+01	0.4136E-02	-.1565E-01	-.2008E-02	0.4099E-02	-.2558E-02	0.3996E-02	-.1329E-03	0.7146E-04	0.6800E+01
0.6900E+01	0.2238E-02	-.2253E-01	-.3771E-02	0.4714E-02	-.1827E-01	0.2002E-02	-.2278E-03	0.1206E-03	0.6900E+01
0.7000E+01	-.3025E-03	-.2837E-01	-.5374E-02	0.5277E-02	-.3113E-01	-.5980E-03	-.1622E-03	0.8498E-04	0.7000E+01
0.7100E+01	-.3369E-02	-.3283E-01	-.6739E-02	0.5709E-02	-.4521E-01	-.3720E-02	-.1388E-03	0.7196E-04	0.7100E+01
0.7200E+01	-.6779E-02	-.3500E-01	-.7770E-02	0.5938E-02	-.5764E-01	-.7167E-02	-.2028E-03	0.1057E-03	0.7200E+01
0.7300E+01	-.1026E-01	-.3387E-01	-.8340E-02	0.5903E-02	-.6815E-01	-.1068E-01	-.1182E-03	0.6055E-04	0.7300E+01
0.7400E+01	-.1345E-01	-.2914E-01	-.8393E-02	0.5563E-02	-.7296E-01	-.1398E-01	-.1059E-03	0.5454E-04	0.7400E+01
0.7500E+01	-.1601E-01	-.2135E-01	-.7912E-02	0.4910E-02	-.6948E-01	-.1644E-01	-.2409E-05	0.4708E-06	0.7500E+01
0.7600E+01	-.1774E-01	-.1303E-01	-.7021E-02	0.3972E-02	-.6383E-01	-.1813E-01	0.9170E-05	-.4315E-05	0.7600E+01
0.7700E+01	-.1870E-01	-.6026E-02	-.5840E-02	0.2794E-02	-.5945E-01	-.1915E-01	-.7549E-05	0.5488E-05	0.7700E+01
0.7800E+01	-.1903E-01	-.5676E-03	-.4557E-02	0.1427E-02	-.5187E-01	-.1954E-01	0.1591E-04	-.5195E-05	0.7800E+01
0.7900E+01	-.1889E-01	0.3360E-02	-.3353E-02	-.7477E-04	-.4581E-01	-.1950E-01	0.2078E-04	-.5776E-05	0.7900E+01
0.8000E+01	-.1841E-01	0.6071E-02	-.2277E-02	-.1660E-02	-.4088E-01	-.1899E-01	0.1356E-04	-.5680E-06	0.8000E+01
0.8100E+01	-.1772E-01	0.7866E-02	-.1337E-02	-.3288E-02	-.3616E-01	-.1832E-01	-.2284E-04	0.2074E-04	0.8100E+01
0.8200E+01	-.1686E-01	0.9163E-02	-.5120E-03	-.4927E-02	-.3222E-01	-.1731E-01	-.3475E-05	0.1157E-04	0.8200E+01
0.8300E+01	-.1590E-01	0.1021E-01	0.2096E-03	-.6552E-02	-.3021E-01	-.1639E-01	-.6866E-05	0.1481E-04	0.8300E+01
0.8400E+01	-.1483E-01	0.1111E-01	0.8320E-03	-.8143E-02	-.2602E-01	-.1525E-01	0.1566E-05	0.1160E-04	0.8400E+01
0.8500E+01	-.1368E-01	0.1191E-01	0.1366E-02	-.9682E-02	-.2352E-01	-.1413E-01	0.2121E-04	0.2480E-05	0.8500E+01
0.8600E+01	-.1246E-01	0.1254E-01	0.1826E-02	-.1115E-01	-.1998E-01	-.1293E-01	-.2739E-04	0.2960E-04	0.8600E+01
0.8700E+01	-.1118E-01	0.1313E-01	0.2239E-02	-.1254E-01	-.1792E-01	-.1160E-01	0.2323E-04	0.3176E-05	0.8700E+01
0.8800E+01	-.9843E-02	0.1359E-01	0.2595E-02	-.1384E-01	-.1429E-01	-.1025E-01	-.1315E-04	0.2317E-04	0.8800E+01
0.8900E+01	-.8469E-02	0.1391E-01	0.2896E-02	-.1503E-01	-.1267E-01	-.8847E-02	0.2464E-05	0.1493E-04	0.8900E+01
0.9000E+01	-.7066E-02	0.1410E-01	0.3142E-02	-.1611E-01	-.8016E-02	-.7391E-02	0.1523E-04	0.8622E-05	0.9000E+01

* * * INTEGRATION SEGMENT 3

TIME	PSI	PSIDOT	BETA	ETA	DELTA	PSI$	N$	Y$	X
0.9000E+01	-.7066E-02	0.1410E-01	0.3142E-02	-.1611E-01	-.8016E-02	-.7391E-02	0.1523E-04	0.8622E-05	0.9000E+01
0.9500E+01	-.1955E-03	0.1296E-01	0.3602E-02	-.1965E-01	0.2239E-02	-.6604E-03	0.2881E-05	0.1560E-04	0.9500E+01
0.1000E+02	0.5551E-02	0.9598E-02	0.3134E-02	-.1998E-01	0.1216E-01	0.5161E-02	0.3556E-05	0.1311E-04	0.1000E+02
0.1050E+02	0.9249E-02	0.5099E-02	0.2089E-02	-.1752E-01	0.1344E-01	0.8904E-02	0.1093E-04	0.6098E-05	0.1050E+02
0.1100E+02	0.1065E-01	0.6978E-03	0.8561E-03	-.1319E-01	0.1237E-01	0.1044E-01	0.8350E-05	0.4309E-05	0.1100E+02
0.1150E+02	0.1013E-01	-.2560E-02	-.2089E-03	-.8071E-02	0.1095E-01	0.1002E-01	0.2869E-04	-.9718E-05	0.1150E+02
0.1200E+02	0.8227E-02	-.4669E-02	-.1005E-02	-.3110E-02	0.5040E-02	0.7991E-02	0.2189E-04	-.8207E-05	0.1200E+02
0.1250E+02	0.5654E-02	-.5425E-02	-.1426E-02	0.1004E-02	0.2540E-02	0.5363E-02	-.2800E-05	0.3761E-05	0.1250E+02
0.1300E+02	0.3038E-02	-.4895E-02	-.1465E-02	0.3906E-02	-.1230E-02	0.2429E-02	0.1342E-05	0.1383E-05	0.1300E+02
0.1350E+02	0.7788E-03	-.3949E-02	-.1304E-02	0.5553E-02	-.3939E-02	0.1488E-03	-.7810E-05	0.6420E-05	0.1350E+02
0.1400E+02	-.8216E-03	-.2353E-02	-.9241E-03	0.6079E-02	-.5029E-02	-.1685E-02	0.4856E-05	0.1558E-06	0.1400E+02
0.1450E+02	-.1682E-02	-.1117E-02	-.5529E-03	0.5799E-02	-.3770E-02	-.2577E-02	-.4531E-04	0.2783E-04	0.1450E+02
0.1500E+02	-.1956E-02	-.1261E-03	-.2215E-03	0.5055E-02	-.4275E-02	-.2700E-02	-.2463E-04	0.1685E-04	0.1500E+02
0.1550E+02	-.1732E-02	0.9296E-03	0.1107E-03	0.4136E-02	-.3436E-02	-.2767E-02	0.3509E-04	-.1407E-04	0.1550E+02
0.1600E+02	-.1253E-02	0.9588E-03	0.2160E-03	0.3302E-02	0.3552E-03	-.1893E-02	-.1281E-05	0.5532E-05	0.1600E+02

Figure 9. SHIPSIM/OPTSIM Printed Output for Example Simulation...Concluded

0.1650E+02	-.7652E-03	0.9939E-03	0.2696E-03	0.2672E-02	-.6680E-03	-.1485E-02	-.1551E-04	0.1330E-04	0.1650E+02
0.1700E+02	-.3081E-03	0.7307E-03	0.2378E-03	0.2278E-02	0.3271E-03	-.6510E-03	-.1047E-04	0.9221E-05	0.1700E+02
0.1750E+02	-.1703E-04	0.5071E-03	0.1820E-03	0.2098E-02	0.7478E-03	-.4750E-03	-.2421E-04	0.1632E-04	0.1750E+02
0.1800E+02	0.2072E-03	0.3851E-03	0.1430E-03	0.2067E-02	0.4726E-03	-.2239E-03	0.3877E-04	-.1839E-04	0.1800E+02
0.1850E+02	0.3636E-03	0.2687E-03	0.1048E-03	0.2150E-02	0.4982E-03	-.1600E-03	-.3985E-04	0.2264E-04	0.1850E+02
0.1900E+02	0.4068E-03	-.3841E-04	0.2397E-04	0.2321E-02	0.6774E-03	-.1209E-03	0.2143E-05	0.5740E-06	0.1900E+02
0.1950E+02	0.3439E-03	-.2235E-03	-.3549E-04	0.2514E-02	0.1623E-02	-.8246E-04	0.1859E-04	-.8188E-05	0.1950E+02
0.2000E+02	0.1601E-03	-.3669E-03	-.9081E-04	0.2682E-02	0.2742E-03	-.1696E-03	0.3591E-04	-.1769E-04	0.2000E+02
0.2050E+02	-.8640E-04	-.6301E-03	-.1612E-03	0.2769E-02	0.6671E-03	-.2735E-03	-.5409E-05	0.4661E-05	0.2050E+02
0.2100E+02	-.3645E-03	-.5289E-03	-.1568E-03	0.2732E-02	-.7471E-03	-.5520E-03	-.1905E-04	0.1165E-04	0.2100E+02
0.2150E+02	-.6032E-03	-.3275E-03	-.1219E-03	0.2564E-02	-.1026E-02	-.7153E-03	0.1023E-04	-.4621E-05	0.2150E+02
0.2200E+02	-.6911E-03	-.1411E-03	-.6671E-04	0.2279E-02	0.5999E-04	-.8063E-03	-.1313E-04	0.7897E-05	0.2200E+02
0.2250E+02	-.7801E-03	-.2836E-04	-.3689E-04	0.1944E-02	-.1202E-02	-.9215E-03	0.2110E-04	-.1026E-04	0.2250E+02
0.2300E+02	-.7566E-03	-.8877E-06	-.1132E-04	0.1565E-02	0.1198E-02	-.8656E-03	-.9969E-06	0.2281E-05	0.2300E+02
0.2350E+02	-.7349E-03	0.1337E-03	0.2264E-04	0.1188E-02	-.1731E-02	-.1002E-02	0.7615E-05	-.1844E-05	0.2350E+02
0.2400E+02	-.6584E-03	0.2546E-03	0.5600E-04	0.8234E-03	-.8557E-03	-.8059E-03	-.1695E-04	0.1100E-04	0.2400E+02
0.2450E+02	-.4625E-03	0.4709E-03	0.1185E-03	0.4920E-03	-.2820E-03	-.4988E-03	0.2237E-04	-.1095E-04	0.2450E+02
0.2500E+02	-.2095E-03	0.4548E-03	0.1338E-03	0.2566E-03	-.8694E-04	-.3305E-03	0.6570E-05	-.2703E-05	0.2500E+02
0.2550E+02	0.3360E-04	0.5318E-03	0.1551E-03	0.1385E-03	0.1097E-02	-.1825E-03	0.4436E-06	0.3878E-06	0.2550E+02
0.2600E+02	0.2682E-03	0.4492E-03	0.1422E-03	0.1412E-03	0.8545E-03	-.1617E-03	0.8462E-06	0.7264E-06	0.2600E+02
0.2650E+02	0.4594E-03	0.1986E-03	0.9134E-04	0.2623E-03	0.2604E-02	0.1218E-03	0.2223E-04	-.1005E-04	0.2650E+02
0.2700E+02	0.4701E-03	-.2577E-04	0.2184E-04	0.4767E-03	-.1078E-02	0.2964E-03	0.9834E-05	-.4389E-05	0.2700E+02
0.2750E+02	0.4220E-03	-.3643E-03	-.6130E-04	0.7077E-03	0.2082E-02	0.5261E-03	-.3700E-04	0.2057E-04	0.2750E+02
0.2800E+02	0.1986E-03	-.5347E-03	-.1267E-03	0.9127E-03	0.1764E-02	0.3868E-03	0.2935E-04	-.1471E-04	0.2800E+02
0.2850E+02	-.1308E-03	-.7301E-03	-.1934E-03	0.1016E-02	0.5319E-03	0.1594E-03	-.3227E-05	0.2277E-05	0.2850E+02
0.2900E+02	-.4381E-03	-.4243E-03	-.1486E-03	0.9590E-03	-.1315E-02	-.4497E-03	-.6932E-05	0.4379E-05	0.2900E+02
0.2950E+02	-.5908E-03	-.1962E-03	-.9015E-04	0.7575E-03	-.1423E-02	-.4872E-03	-.1338E-04	0.6921E-05	0.2950E+02
0.3000E+02	-.5969E-03	0.6398E-04	-.1388E-04	0.4766E-03	-.1390E-03	-.3985E-03	-.1504E-04	0.6954E-05	0.3000E+02

```
PDS: PLOT DESCRIPTION GENERATION BEGINS ***


*** END OF FILE ENCOUNTERED ON 4.
#EXECUTION TERMINATED
#



#DISPLAY COST
#COST =    $3.07, TERM,NORMAL,UNIV
#
#RUN *CCQUEUE PAR=-PLOTFILE
#EXECUTION BEGINS
**'-PLOTFILE        ' NOT A PERMANENT FILE.  QUEUEING ABORTED
#EXECUTION TERMINATED
#COPY  -PLOTFILE PLOTFILE4
#RUN *CCQUEUE
#EXECUTION BEGINS
 ENTER PLOT REQUEST:
 PLOTFILE4
   3 PLOTS; PLOTTING REQUIRES    670 SEC. AND    39 IN.;    $1.86
 OK?
 OK
 PLOT ASSIGNED RECEIPT # 510132.
 ENTER PLOT REQUEST:

 EOF
#EXECUTION TERMINATED
#
```

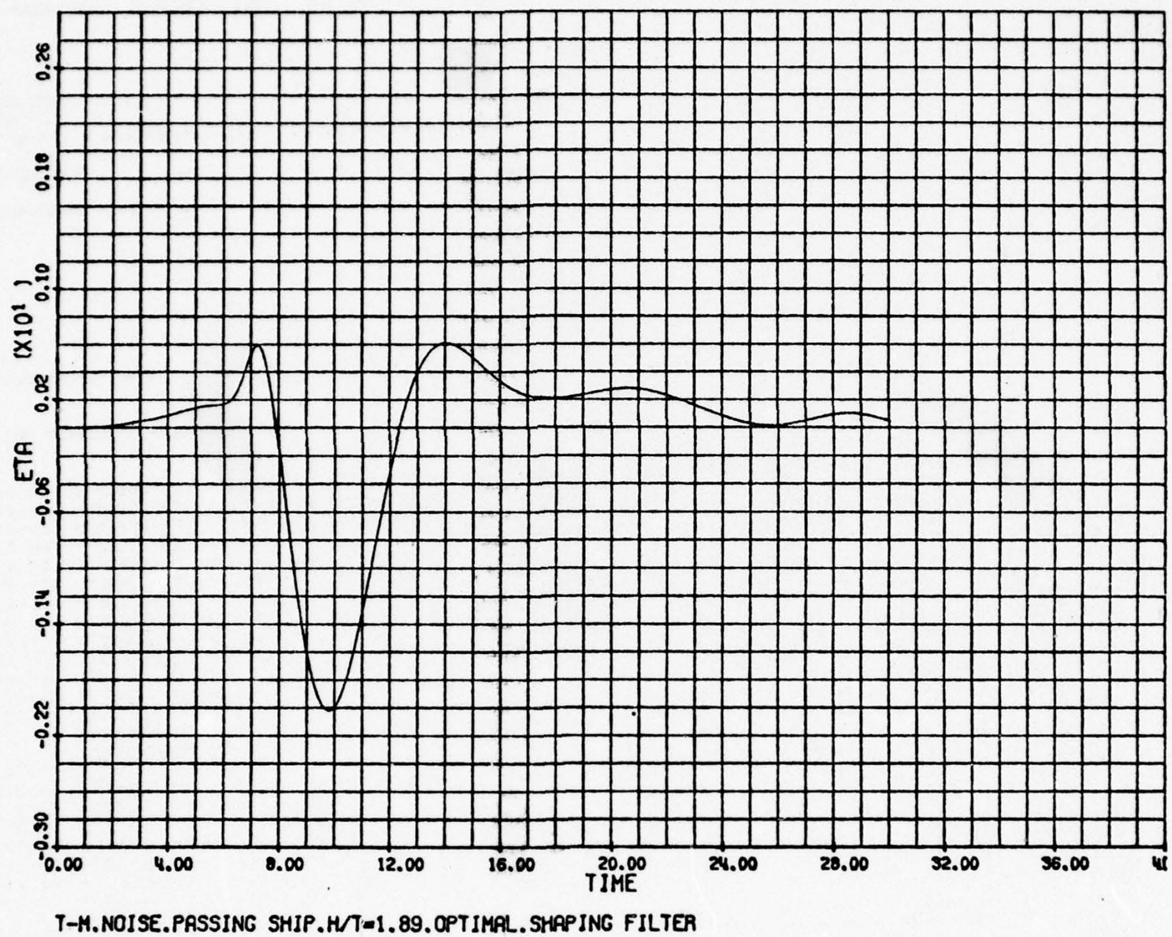


Figure 10. Plot of Lateral Offset η' from Example Simulation

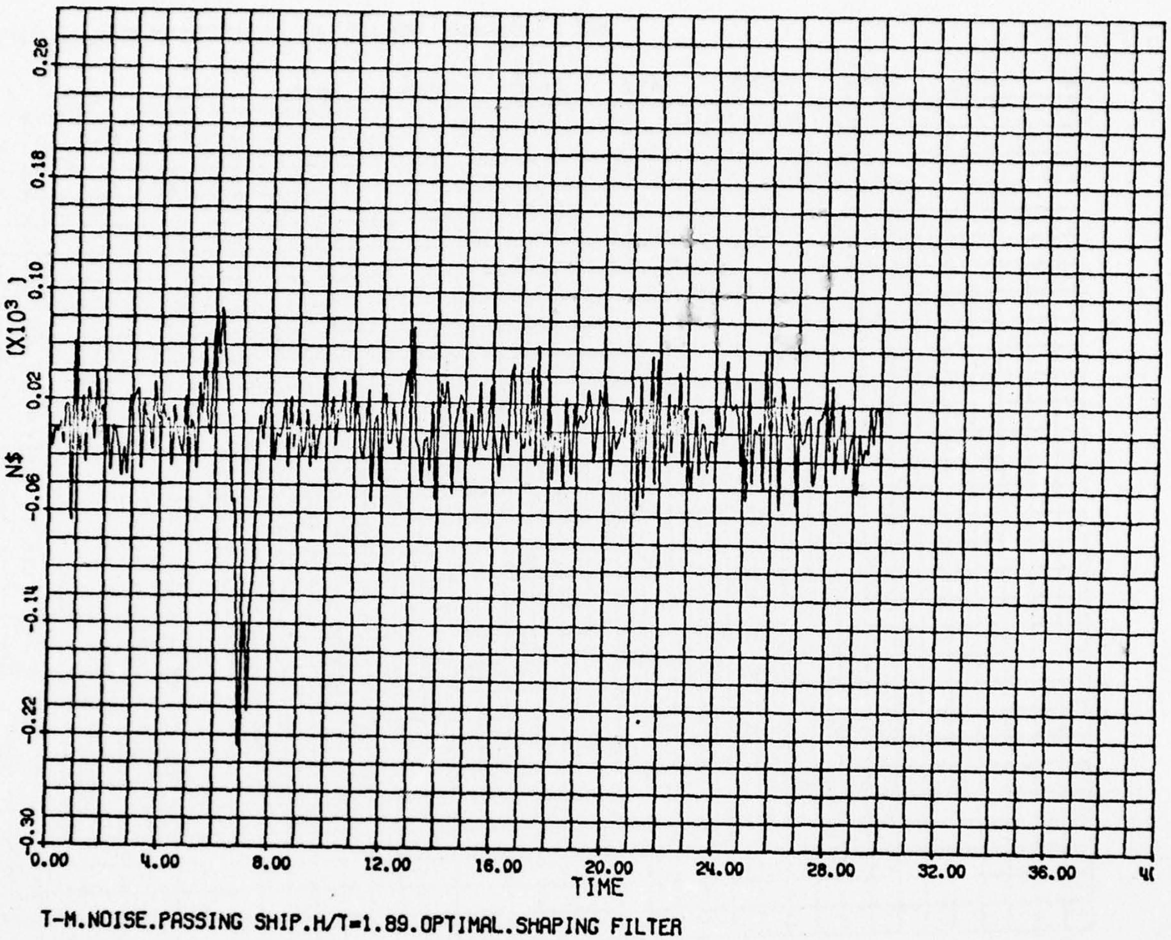


Figure 11. Plot of Yawing Moment Estimate $\hat{N}'$ from Example Simulation

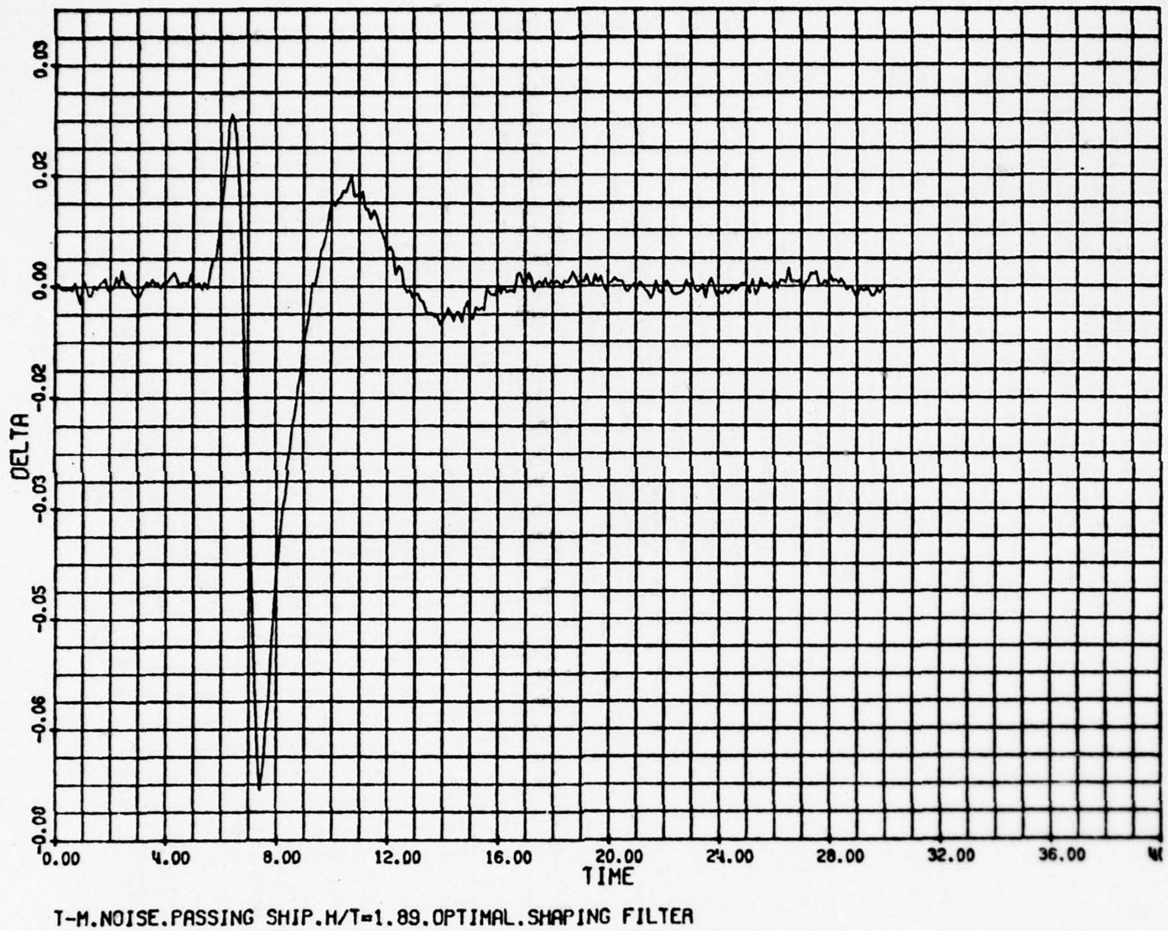


Figure 12. Plot of Rudder Angle δ' from Example Simulation

VI. Closure

The SHIPSIM/OPTSIM program used in conjunction with the OPTSYS program provides a useful and efficient design tool for the development of optimal stochastic control systems for stationary, linear systems. The SHIPSIM program used alone provides a useful continuous system simulation program with a wide range of applications. The program provides a compromise between writing simulation programs from scratch using existing integration subroutines and using higher-level, problem-oriented simulation languages such as IBM's CSMP. Both these programs can be utilized with a minimum of user programming and data preparation.

VII. REFERENCES

1. Parsons, M.G. and Cuong, H.T., "Optimal Stochastic Path Control of Ships in Shallow Water," ONR Report ONR-CR215-249-2F, 15 August 1977.

2. Bryson, A.E. Jr., and Hall, W.E. Jr., "Optimal Control and Filter Synthesis by Eigenvector Decomposition," Stanford Univ. Guidance and Control Lab., Report SUDAAR #436, Nov., 1971.

3. Bryson, A.E. Jr. and Ho, Y.C., Applied Optimal Control, Blaisdell Publ., Waltham, Mass., 1969.

4. Bryson, A.E., Jr., "Control Theory for Random Systems," Proceedings of the Thirteenth International Congress of Theoretical and Applied Mechanics, Moscow, August, 1972.

5. Gelb, A. (ed.), Applied Optimal Estimation, The M.I.T. Press, Cambridge, Mass., 1974.

6. MacFarlane, A.G.J., "An Eigenvector Solution of the Optimal Linear Regulator Problem," Journal of Electronics and Control, Vol. 14, No. 6, June, 1963, pp. 643-654.

7. Potter, J.E., "Matrix Quadratic Solutions," SIAM Journal of Applied Mathematics, Vol. 14, No. 3, May, 1966, pp. 496-501.

8. Parsons, M.G., "Optimal Control of Linear Systems," University of Michigan, Department of Naval Architecture and Marine Engineering unpublished notes, September, 1976.

9. Severance, R.W., Jr., "Optimum Filtering and Smoothing of Buoy Wave Data," Journal of Hydronautics, Vol. 9, No. 2, April, 1975, pp. 69-74.

10. Parsons, M.G., "Throttle Controls for Stopping Reversible Steam Turbine Driven Ships with Minimum Head Reach," Proceedings of the 1974 Summer Computer Simulation Conference, July 9-11, 1974.

11. Parsons, M.G. and Altemos, E.A., "The Effect of Shaft Brakes on Emergency Stopping Head Reach with Reversible Reduction Gear Propulsion Systems," Proceedings of the 1975 Summer Computer Simulation Conference, July 21-23, 1975.

12. System/360 Continuous System Modeling Program: User's Manual Program Number 360A-CS-16X, IBM Application Program GH20-0367-4, Fifth Edition, 1972.

13. Parsons, M.G. , "Optimization Methods for Use in Computer-Aided Ship Design," Proceedings of the SNAME 1975 Ship Technology and Research (STAR) Symposium, August 26-29, 1975.

14. Fox, L., Numerical Solution of Ordinary and Partial Differential Equations, 1962, p. 24.

15. Lukehart, P.M., "Kutta-Merson, Algorithm 218," Comm. ACM, Vol. 6, No. 12, Dec. 1963, p. 737.

16. Fronczak, E.J., "Plot Description System," MTS, The Michigan Terminal System, Vol. 11, Third Edition, April, 1971.

17. System/360 Scientific Subroutine Package (360A-CM-03X) Version III Programmer's Manual, IBM Application Program, H20-0205-3, Fourth Edition, 1968.

18. Fujino, M., "Studies on Manoeuvrability of Ships in Restricted Waters," Selected Papers from the Journal of the Society of Naval Architects of Japan, Vol. 4, 1970, pp. 157-184.

19. Newton, R.N., "Interaction Effects Between Ships Close Aboard in Deep Water," David Taylor Model Basin, Report 1461, 1960.

Appendix A: User's Documentation for SHIPSIM

UNIVERSITY OF MICHIGAN

DEPARTMENT OF NAVAL ARCHITECTURE AND MARINE ENGINEERING Rev. 2 5/27/77

IDENTIFICATION: SHIPSIM Program

PROGRAMMER: Ass't. Prof. Michael G. Parsons and J.E. Greenblatt of the University of Michigan, Department of Naval Architecture and Marine Engineering, under ONR Contract N00014-76-C-0751.

PURPOSE: This continuous system simulation program integrates a system of up to 25 first-order differential equations. Flexible input, output, and integration control is provided. Up to five auxiliary quantities may be computed and displayed with the integration output. The system definition and auxiliary calculations are handled through user-supplied subroutines. Integration methods available are a fixed step-size Euler or rectangular integration and a fourth-order, variable step-size Kutta-Merson integration with optional absolute and relative error control. Integration method and integration and output control parameters can be changed at up to four user-specified points within the integration. The integration can be terminated based on the value of the independent variable or based on the value of one of the integrated dependent variables.

METHOD:

References:

1. Fox, L., Numerical Solution of Ordinary and Partial Differential Equations, 1962, p. 24.
2. MTS Volume 11: Plot Description System, University of Michigan Computing Center, 1971.
3. IBM System/360 and System/370 FORTRAN IV Language, IBM Manual Number GC28-6515-10, May, 1974.

The only computational function of SHIPSIM is to integrate a set of NEQ(≤25) coupled, first-order differential equations,

$$\underset{NEQx1}{\dot{\underline{Y}}(t)} = \underline{f}(\underline{Y},t) \quad , \quad \underline{Y}(t_o) = \underline{Y}_o \quad ,$$

from an initial condition through time and to perform a set of NAC(≤5) auxiliary calculations,

$$\underset{NACx1}{\underline{z}(t)} = \underline{g}(\underline{Y},t) \quad ,$$

at each program output point. The integration can be performed using a rectangular or Euler integration or using a Kutta-Merson integration technique.

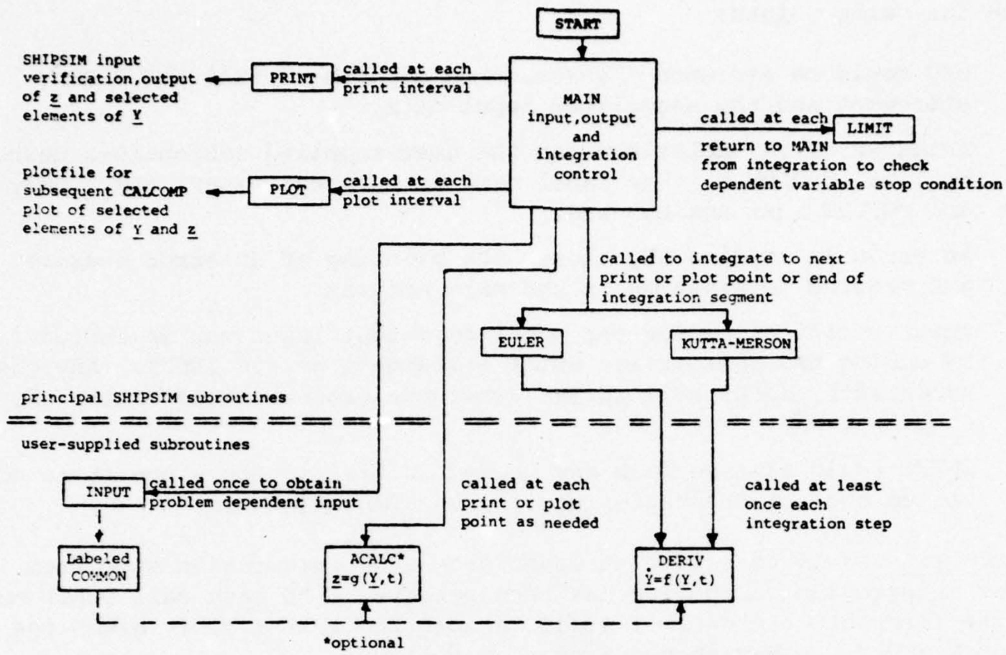
Euler integration gives the value of $\underline{Y}$ at time $t+\Delta t$ as,

$$\underline{Y}(t+\Delta t) = \underline{Y}(t) + \dot{\underline{Y}}(t)*\Delta t \quad ;$$

i.e., it evaluates the derivatives once at time t and assumes them contant over the integration step Δt. Euler integration is thus very simple and efficient. For processes with generally small and/or smooth variations in $\dot{\underline{Y}}$ it can yield acceptable answers when a suitably small integration step-size, Δt, is used. *When the step-size is too large it is possible for the integration results to completely diverge from the correct results.* Euler integration can also successfully handle discrete changes in $\dot{\underline{Y}}$ when Δt is kept small. If $\dot{\underline{Y}}$ experiences large and rapid changes in magnitude, the acceptable value of Δt may be so small that excessive CPU time will be needed to complete the integration. The integration step-size used in a region where $\dot{\underline{Y}}$ changes rapidly would be very wasteful if also used in regions where $\dot{\underline{Y}}$ changes more slowly. For this reason (and to facilitate variations in output), SHIPSIM allows a specific integration run to be specified with up to five integration segments each with a separate integration method and/or step-size. Euler integration has the disadvantage that the error of the integration results is unknown. It is therefore essential to perform test integration runs with a reduced step-size to verify that the results are acceptably accurate.

Kutta-Merson integration[1] is much more complex than Euler integration but provides a dynamically varying integration step-size which is automatically doubled or halved as necessary to produce guaranteed, user-specified absolute and/or relative error in the results. This step-size will be short where $\dot{\underline{Y}}$ changes rapidly and much larger where $\dot{\underline{Y}}$ changes more slowly. There is much more computational overhead than with Euler integration but guaranteed error is provided and the integration step-size is never any shorter than necessary. Improved integration cost is thus possible in some cases. A major problem with Kutta-Merson integration is that it may be very difficult to meet the specified error limits (particularly at points where $\dot{\underline{Y}}$ is discontinuous in value or slope) without shortening the integration step-size to the point where excessive CPU time is used. To protect against excessive reduction in step-size without user interaction to relax error specifications, SHIPSIM includes capability to limit the number of times the initially specified step-size will be automatically halved (cut). If this number of cuts is exceeded, the integration is terminated and an error message is printed. Kutta-Merson integration is able to predict the integration error by evaluating the derivatives at five points in each integration step as compared to Euler integration which evaluates the derivatives only once for each integration step.

<u>PROGRAM DESCRIPTION</u>: The basic organization of SHIPSIM is shown in the following macro flow chart which includes only the principal SHIPSIM subroutines:



The MAIN program regulates the integration and controls output processing. User-supplied, double-precision subroutines, INPUT, DERIV, and ACALC define the system and any auxiliary calculations of interest. Subrouting PRINT writes the requested simulation results. *Subroutine PLOT prepares* a plotfile which can be used for subsequent generation of CALCOMP plots of selected simulation results. Subroutine LIMIT checks integration progress and terminates the run if an element of Y reaches a user-prescribed value. Subroutines EULER and KUTTA-MERSON conduct the requested integration.

USER-SUPPLIED SUBROUTINES: The following user-supplied, double-precision, subroutines should be compiled and then loaded into an object file(s) which can be referenced on the MTS RUN command.

Subroutine INPUT is called once at the beginning of a simulation run to read any needed problem dependent input on logical I/O device 5 and to transmit this data to the other user-supplied subroutines. It must also return the number of equations to be integrated (NEQ) to the main program. Input should appear as follows:

```
      SUBROUTINE INPUT(NEQ,*)
      COMMON/MODEL/...[data transfer to DERIV and/or ACALC]
      READ(5,1ØØ,ERR=2ØØ)NEQ
1ØØ   FORMAT(I2)
        :
        code to read any problem dependent input
        :
      RETURN
2ØØ   WRITE(6,21Ø)
21Ø   FORMAT('-ERROR IN READING NEQ')
      RETURN 1
      END
```

Note the following points:

1. NEQ could be assigned a value, thus avoiding a READ and FORMAT statement and the associated input data.
2. Transfer of variables between the user-supplied subroutines must be by labeled COMMON. The label name is optional except that COM1, COM2, and OUTPUT must not be used.
3. An error in reading NEQ causes the printing of an error message and program termination in the main program.
4. Input verification for the model dependent input may be included by coding the appropriate WRITE statements within INPUT. Any output produced by INPUT will appear after the program title in the SHIPSIM output stream.
5. INPUT could also perform any needed initialization since it is only called once at the beginning of each simulation run.

Subroutine DERIV is called at least once each integration step from whichever integration subroutine has been selected. At each call DERIV must return the first NEQ elements of d$\underline{Y}$/dT through the vector YDOT, given the values of T and $\underline{Y}$. DERIV should appear as follows:

```
SUBROUTINE DERIV(T,Y,YDOT)
REAL*8 T,Y(25), YDOT(25)
COMMON/MODEL/...[data from INPUT]
   .
   .
   .
 code to define first NEQ elements of YDOT
   .
   .
   .
RETURN
END
```

Subroutine ACALC is an optional subroutine which is called as needed at each print or plot point to calculate up to five non-integrated auxiliary variables,

$$\underset{NACx1}{\underline{z}} = \underline{z}(\underline{Y},T) \ .$$

This subroutine is called only if a number of auxiliary calculations are specified by the input variable NAC>Ø. All NAC elements of $\underline{z}$ will automatically appear in the SHIPSIM printed output. If NAC=0, the user can call a dummy subroutine as part of the MTS RUN command as described below and there is no need to write ACALC. When NAC>0, ACALC should appear as follows:

```
SUBROUTINE ACALC(T,Y,Z,NAC)
REAL*8 T,Y(25), z(5)
COMMON/MODEL/...[Parameters from INPUT]
   .
   .
   .
  code to calculate first NAC elements of z
   .
   .
   .
RETURN
END
```

OUTPUT Options: Three output options are available. Printed output is written on logical I/O channel 6. Format may be selected by the input variable IOUT. If IOUT=1, up to nine quantities plus the time are printed in tabular form across 120 columns, for each integration output point. These nine quantities are the NAC elements of z preceeded by up to any (9-NAC) elements of Y. The elements of Y must be specified by the input vector CPRINT. If IOUT=2, output is given in a vector format. All NEQ elements of Y and the NAC elements of z are printed.

SHIPSIM will also generate a plot file for MTS graphic post-processing if the input variable PLN (number of plots) is greater than zero. Use is made of the MTS plotting subroutine library contained in *PLOTSYS.[2] Variables for which plots are desired must be specified by the input variable CPLOT. The plot time increment PLD should be selected to give no more than 300 data points per plot.

USER'S INSTRUCTIONS: As described above, the MAIN program reads various simulation specification and control input in addition to the input which is brought in by the user-supplied subroutine INPUT. All MAIN program input is read on logical I/O device 4. Initial runs must be specified by a complete set of fixed-format input data as specified below. Subsequent runs may be specified either by inputting a completely new set of fixed-format data or by changing only specific values in the data for the previous run by using free-format NAMELIST data sets.

The data sets for multiple simulation runs would be read in the following sequence:

1. a complete set of user's problem dependent data for subroutine INPUT on I/O channel 5.
2. a complete set of SHIPSIM data on I/O channel 4.
3. SHIPSIM input NEXT on I/O channel 4.
4. if NEXT=1, SHIPSIM NAMELIST data; existing model dependent data will be reused
   if NEXT=2, SHIPSIM NAMELIST data followed by new model dependent INPUT data
   if NEXT=3, SHIPSIM fixed-format input data; existing model dependent data will be reused
   if NEXT=4, new model dependent INPUT data followed by SHIPSIM fixed-format data
   if NEXT=5, new model dependent INPUT data only
5. a new value of NEXT for third run, etc.

Fixed-format SHIPSIM data consists of the following in the specified sequence including only those required:

Record Type	Variables	Format	Comments
1	NIS NAC IOUT PLN SF*	I5 I5 I5 I5 D1Ø.4	Number of integration segments Ø<NIS≤5 Number of auxiliary quantities Ø≤NAC≤5 =1 for tabular output =2 for vectorial output Number of plots desired Ø≤PLN≤9 Scale factor for plots. With SF=1.Ø, plots are 8-1/2" x 11". Ø<SF≤1.Ø omit if PLN=Ø
2	TITLE	9A8	Any 72 character user's title for printout and plot labeling
3	YLABLE	8(A8,2X) repeated as required	Labels for NEQ elements of $\underline{Y}$
4	YØ	7D1Ø.4 repeated as required	Initial values for $\underline{Y}$.
5	YTEST YTERM*	A8 D1Ø.4	Name of Y variable to be tested for limiting value. The first time the limiting value is reached from either direction the run stops. If no limiting value is desired enter "NONE____" Limiting value of YTEST. May be omitted if NONE is entered for YTEST
6*	ZLABLE	(A8,2X)	Labels for NAC elements of $\underline{z}$. This record should be omitted if NAC=Ø.
7*	CPRINT	9(A8)	Names of Y variables for which printed tabular output is desired. This record should be omitted if IOUT=2
8*	CPLOT	7(A8,2X)	Names of $\underline{Y}$ and $\underline{Z}$ variables for which plots are desired. The number of names must equal PLN. This record should be omitted if PLN=Ø.

Record Type	Variables	Format	Comment
9 repeated NIS times defining each integration segment in succession. Each input variable is a vector NIS long.	METHOD	I5	Integration method: =1 for Euler =2 for Kutta-Merson
	TF	D1Ø.2	Termination time for the integration segment.
	FIRSTP	D1Ø.2	Initial integration step size
	PRD	D5.2	Time increment at which printed output is desired
	PLD	D5.2	Time increment between plotted points
	EPS*	D1Ø.6	Relative error limit for Kutta-Merson integration. Omit if METHOD=1
	AB*	D1Ø.6	Absolute error limit for Kutta-Merson integration. Omit if METHOD=1
	NCUTS*	I5	Number of times the integration step size may be halved during Kutta-Merson integration. A value of 20 is suggested. Omit if METHOD=1.
10*	NEXT	I1	input switch for next run. Omit if no further runs are desired. NEXT=1, SHIPSIM NAMELIST data on I/O channel 4; existing model dependent data will be reused NEXT=2, SHIPSIM NAMELIST data on I/O channel 4; new model dependent data for user's INPUT subroutine on I/O channel 5. NEXT=3, SHIPSIM fixed-format input on I/O channel 4; existing model dependent data will be reused. NEXT=4, new model dependent data for user's INPUT subroutine on I/O channel 5; SHIPSIM fixed-format data on I/O channel 4 NEXT=5, new model dependent data for user's INPUT subroutine on I/O channel 5; existing SHIPSIM data will be reused. INPUT data only.

* These records or variables may be omitted under the specified conditions.

For runs where the NAMELIST input option is specified, (NEXT=1 or 2) the data should consist of one (or more) lines of the form,

```
column  2
        &DATA TF(2)=2Ø., PRD(2)=1.Ø,
        METHOD(2)=2, &END
```

If all changes can be made on one line MTS permits the omission of the &DATA and &END delimiters. The user should review the use of NAMELIST in the IBM FORTRAN IV language manual.[3] Note that any variable input as an alpha-numeric string must be enclosed in apostrophes, as:

```
... CPLOT(1) ='VELOCITY', ...
```

Both the Euler and Kutta-Merson integration subroutines are called several times within each specified integration segment. Each call specifies an integration interval equal to the smaller of the printing interval PRD and the plotting interval PLD. If the smaller of PRD and PLD is not an integer multiple of FIRSTP the output may be shifted slightly from that expected when Euler integration is used. Further, if the final time for this segment (TF) is not an integer multiple of the smaller of PRD and PLD, it also will be shifted slightly by the program. The larger of PLD and PRD must be an integer multiple of the small of these two quantities. For best results, the smaller of PLD and PRD should be an integer multiple of FIRSTP when Euler integration is specified. With either integration method, TF and the larger of PLD and PRD should be an integer multiple of the smaller of PLD and PRD. The following specification would thus be consistent: FIRSTP=0.01, PRD=1.0, PLD=5.0, TF=100.0.

MTS RUN INFORMATION: Object code for SHIPSIM is currently in file SGTA:SHIPSIM.O (subject to change, see Prof. Michael Parsons). If no auxiliary calculations are required object code for a dummy ACALC subroutine is available in file SGTA:ACALC.D. A run using this dummy subroutine would be initiated with the command:

```
$RUN SGTA:SHIPSIM.O+SGTA:ACALC.D+*PLOTSYS+(user's INPUT and
DERIV object file) 4=(user's SHIPSIM data file) 5=(user's
INPUT data file) 9=(user's plotfile)
```

If no plots are required (PLN=Ø), device 9 may be assigned to *DUMMY*. If plots are required each one will require approximately three pages of file space and cost approximately $.65. If a CALCOMP plot is desired the following commands could follow.

```
$ PERMIT (User's plotfile) READ OTHERS
$ RUN   *CCQUEUE  PAR=(User's plotfile)
```

For a description of *CCQUEUE see MTS volume 11.[2] The plotfile can be examined on an interactive graphics terminal by issuing the following command:

```
$RUN *PLOTSEE
```

and then responding with the user's plotfile name.

If a large amount of printed output is anticipated, the SHIPSIM printed output channel 6 may be assigned to the high-speed line printer *PRINT* or to a user file for later copying to *PRINT*.

EXAMPLE PROBLEM: The following problem illustrates the use of SHIPSIM to simulate the straight line crash stop of a gas turbine powered escort vessel with a controllable pitch propeller. Suppose we wish to investigate the effect of a throttle application ramp on engine and shaft torque and stopping distance.

Our simplified ship model has the following characteristics:

M = 4000 L.T. = 8,960,000 lbm.

M_o (added mass) = 8%

J_e (engine inertia) = 108,864 lbf-ft-sec^2

$J_p + J_o$ (propeller inertia plus added inertia) = 50,000 lbf-ft-sec^2

V_o = steady-state ahead speed
= 25 knots = 42.225 ft/sec

N_{po} = steady-state shaft speed
= 4.16667 sec^{-1}

w' = wake fraction = .06

t' = thrust deduction = .10

η_r = relative rotative efficiency = 1.0

D = propeller diameter = 13.37 ft.

DAR = developed area ratio = .70

P = pitch/diameter, $-1.0 \le P \le 1.4$

$\dot{P}$ = pitch/diameter change rate = 0.1 sec^{-1}

Machinery characteristics are:

drive train frictional torque = -3% of Q_e

TH = throttle, $.18 \le TH \le 1.0$

Engine torque characteristics are given by

$$\frac{Q_e}{Q_{eo}} = (-.2928\ TH^2 + .16260\ TH - .01332)\left(2-\frac{N_e}{N_{eo}}\right) - 0.20$$

The normalized equations of motion are:

$$\dot{\tilde{x}} = (v_o/x_o)\ \tilde{v} \quad ,$$

$$\dot{\tilde{v}} = \frac{T_{po}g_c}{(M+M_o)v_o} \quad [\tilde{T}_p-\tilde{v}^2] \quad ,$$

$$\dot{N} = \frac{Q_{eo}}{2\pi(J_e+J_p+J_o)N_o} [\tilde{Q}_e+\tilde{Q}_p-\tilde{Q}_F] ,$$

where,

v_o = normalizing value of ship velocity
= 42.225 ft/sec

x_o = normalizing value of head reach
= 1000 ft.

T_{po} = normalizing value of propeller thrust
= 180,055.7 lbf

Q_{eo} = normalizing value of engine torque
= 625,580.6 ft. lbf.

N_o = normalizing value of shaft revolution rate
= 4.1667 sec^{-1}

$\tilde{x}$ = normalized ship position

$\tilde{v}$ = normalized ship velocity

$\tilde{N}$ = normalized shaft speed

$\tilde{T}_p$ = normalized propeller thrust

$\tilde{Q}_e$ = normalized engine torque

$\tilde{Q}_p$ = normalized propeller torque

$\tilde{Q}_f$ = normalized frictional torque

The user-supplied subroutines INPUT, DERIV, and ACALC are shown below. These subroutines were compiled and loaded into file PS8.0. Subroutine CRPROP is an external subroutine which returns propeller thrust, torque, and efficiency given speed of advance, rotation rate, wake characteristics and propeller geometry. Subroutine CRPROP will not be listed here.

```
-     SUBROUTINE INPUT(NEQ,*)
      IMPLICIT REAL*8 (A-H,O-$)
      COMMON/MODEL/PZ,S
      READ(5,100,ERR=150) PZ, S
100   FORMAT(2D15.9)
      WRITE(6,110) PZ, S
110   FORMAT('-INPUT VERIFICATION: STEADY STATE PITCH= ',
     *   F6.3, ' D(THROTTLE)/DT= ', F6.4)
      NEQ=3
      RETURN
150   WRITE(6,160)
160   FORMAT(' ERROR IN READING PZ OR S ')
      RETURN1
      END
C
C
      SUBROUTINE DERIV(T,Y,YDOT)
      IMPLICIT REAL*8 (A-H,O-$)
      COMMON/MODEL/PZ,S
      DIMENSION Y(25),YDOT(25)
C
C     CALCULATE PITCH FOR CRPROP
C
      IF (T .LT. 5.) P=PZ
      IF (5. .LE. T .AND. T .LT. 26.5) P=PZ-0.1*(T-5.)
      IF (26.5 .LE. T) P=-1.0
C
```

```
C     OBTAIN NORMALIZED PROP THRUST AND TORQUE
C
      CALL CRPROP(Y(2),Y(3),P,.7D00,0,0,0,TPN,QPN,EFFO,
     *   A,B,C,D,E,F,1.99D00,1.337D01,0.9D00,0.94D00,
     *   6.25361D05,1.80000D05,4.2225D01,4.166666667D00)
C
C     CALCULATE ENGINE THRUST (TEN) AND THROTTLE (TH)
C
      IF (T .LT. 5.) TH=1.0
      IF (5. LE. T .AND. T .LT. 16.5) TH=.18
      IF (16.5 .LE. T) TH=0.18+S*(T-16.5)
      IF (TH .GT. 1.0) TH=1.0
      QEN=(-.2928*TH**2+1.626*TH-0.1332)*(2.0-Y(3))-0.2
C
      YDOT(1)=0.042225*Y(2)
      YDOT(2)=0.0141849*(TPN-Y(2)**2)
      YDOT(3)=0.150267054*(QPN+0.97*QEN)
      RETURN
      END
C
C
      SUBROUTINE ACALC(T,Y,Z,NAC)
      IMPLICIT REAL*8 (A-H,O-$)
      COMMON/MODEL/PZ,S
      DIMENSION Z(5),Y(25)
C
C     CALCULATE PITCH FOR CRPROP
C
      IF (T .LT. 5.) P=PZ
      IF (5. .LE. T .AND. T .LT. 26.5) P=PZ-0.1*(T-5.)
      IF (26.5 .LE. T) P=-1.0
C
C     OBTAIN NORMALIZED PROP THRUST AND TORQUE
C
      CALL CRPROP(Y(2),Y(3),P,.7D00,0,0,0,TPN,QPN,EFFO,
     *   A,B,C,D,E,F,1.99D00,1.337D01,0.9D00,0.94D00,
     *   6.25361D05,1.80000D05,4.2225D01,4.166666667D00)
C
C     CALCULATE ENGINE THRUST (TEN) AND THROTTLE (TH)
C
      IF (T .LT. 5.) TH=1.0
      IF (5. LE. T .AND. T .LT. 16.5) TH=.18
      IF (16.5 .LE. T) TH=0.18+S*(T-16.5)
      IF (TH .GT. 1.0) TH=1.0
      QEN=(-.2928*TH**2+1.626*TH-0.1332)*(2.0-Y(3))-0.2
      Z(1)=P
      Z(2)=TH
      Z(3)=QEN
      Z(4)=QPN
      RETURN
      END
```

The data for SHIPSIM, shown below, calls for plotting of shaft speed and throttle in the first run. For the second run (NEXT=2) SHIPSIM NAMELIST input alters the title, eliminates plot generation, and changes IOUT to 2 to produce the vectorial form output. This data was loaded into file D4.

```
   2    4    1    2   1.0
CRASH STOP SIMULATION: IDLE TO FULL THROTTLE IN 20 SECONDS
ADVANCE   VELOCITY  SHAFT N
0.0        1.0        1.0
VELOCITY  -0.001
PITCH/D    THROTTLE  ENGINE Q  PROP Q
ADVANCE VELOCITYSHAFT N
SHAFT N   THROTTLE
    1  30.      0.1       1.0  1.0
    1  60.      1.0       1.0  1.0
2
 &DATA TITLE='CRASH STOP SIMULATION: IDLE TO FULL THROTTLE IN 10 SEC.',
      PLN=0, IOUT=2,    &END
```

Data for subroutine INPUT consists of two lines, one for each run. The first value (PZ) is the steady-state ahead pitch. The second (S) is the slope of the throttle return ramp. This data was loaded into file D5.

```
1.14778      0.041D00
1.14778      0.082D00
```

The program output follows.

```
#RUN SHIPSIM.O+PB8.O+CRPROP.O+*PLOTSYS 4=D4 5=D5 9=PLOTFILE

#EXECUTION BEGINS



UNIVERSITY OF MICHIGAN DEPARTMENT OF NAVAL ARCHITECTURE AND MARINE ENGINEERING

SHIPSIM CONTINUOUS SYSTEMS SIMULATION PROGRAM


INPUT VERIFICATION: STEADY STATE PITCH=  1.148 D(THROTTLE)/DT= 0.0410

CRASH STOP SIMULATION: IDLE TO FULL THROTTLE IN 20 SECONDS

   * * * VARIABLES AND INITIAL VALUES:

         ADVANCE  = 0.0        VELOCITY = 0.100E+01   SHAFT N  = 0.100E+01

   * * * AUXILIARY VARIABLES:

         PITCH/D   THROTTLE  ENGINE Q  PROP Q

   * * * VARIABLES TO BE PRINTED:      ADVANCE   VELOCITY  SHAFT N

   * * * VARIABLES TO BE PLOTTED:      SHAFT N   THROTTLE

   * * * LIMITING VALUE OF VELOCITY IS -.1000E-02

   * * * INTEGRATION CONTROL PARAMETERS:

         SEGMENT  METHOD   TF          PRD         FLD         FIRSTP      EPS        AB        NCUTS

            1     EULER  0.3000E+02  0.1000E+01  0.1000E+01  0.1000E+00
            2     EULER  0.6000E+02  0.1000E+01  0.1000E+01  0.1000E+01



   * * * INTEGRATION SEGMENT 1

    TIME        ADVANCE      VELOCITY    SHAFT N     PITCH/D     THROTTLE    ENGINE Q    PROP Q
0.0          0.0          0.1000E+01  0.1000E+01  0.1148E+01  0.1000E+01  0.1000E+01  -.9697E+00
0.1000E+01   0.4222E-01   0.1000E+01  0.1000E+01  0.1148E+01  0.1000E+01  0.1000E+01  -.9698E+00
0.2000E+01   0.8445E-01   0.1000E+01  0.1000E+01  0.1148E+01  0.1000E+01  0.9999E+00  -.9699E+00
```

0.3000E+01	0.1262E+00	0.1000E+01	0.1000E+01	0.1148E+01	0.1000E+01	0.9999E+00	-.9699E+00
0.4000E+01	0.1689E+00	0.1000E+01	0.1000E+01	0.1148E+01	0.1000E+01	0.9999E+00	-.9699E+00
0.5000E+01	0.2111E+00	0.1000E+01	0.1000E+01	0.1148E+01	0.1000E+01	0.9999E+00	-.9699E+00
0.6000E+01	0.2533E+00	0.9962E+00	0.9014E+00	0.1048E+01	0.1800E+00	-.3521E-01	-.4491E+00
0.7000E+01	0.2952E+00	0.9863E+00	0.8486E+00	0.9478E+00	0.1800E+00	-.2729E-01	-.1966E+00
0.8000E+01	0.3366E+00	0.9734E+00	0.8254E+00	0.8478E+00	0.1800E+00	-.2382E-01	-.6132E-01
0.9000E+01	0.3774E+00	0.9587E+00	0.8187E+00	0.7478E+00	0.1800E+00	-.2281E-01	0.1830E-01
0.1000E+02	0.4176E+00	0.9427E+00	0.8216E+00	0.6478E+00	0.1800E+00	-.2324E-01	0.6486E-01
0.1100E+02	0.4571E+00	0.9256E+00	0.8296E+00	0.5478E+00	0.1800E+00	-.2445E-01	0.8727E-01
0.1200E+02	0.4956E+00	0.9073E+00	0.8394E+00	0.4478E+00	0.1800E+00	-.2592E-01	0.8865E-01
0.1300E+02	0.5338E+00	0.8878E+00	0.8479E+00	0.3478E+00	0.1800E+00	-.2719E-01	0.6960E-01
0.1400E+02	0.5709E+00	0.8670E+00	0.8519E+00	0.2478E+00	0.1800E+00	-.2779E-01	0.2992E-01
0.1500E+02	0.6071E+00	0.8447E+00	0.8485E+00	0.1478E+00	0.1800E+00	-.2729E-01	-.2993E-01
0.1600E+02	0.6423E+00	0.8209E+00	0.8351E+00	0.4778E-01	0.1800E+00	-.2528E-01	-.1076E+00
0.1700E+02	0.6765E+00	0.7958E+00	0.8106E+00	-.5222E-01	0.2005E+00	0.1533E-01	-.1980E+00
0.1800E+02	0.7096E+00	0.7694E+00	0.7817E+00	-.1522E+00	0.2415E+00	0.9533E-01	-.2968E+00
0.1900E+02	0.7416E+00	0.7423E+00	0.7495E+00	-.2522E+00	0.2825E+00	0.1786E+00	-.3986E+00
0.2000E+02	0.7724E+00	0.7148E+00	0.7145E+00	-.3522E+00	0.3235E+00	0.2656E+00	-.4978E+00
0.2100E+02	0.8021E+00	0.6872E+00	0.6780E+00	-.4522E+00	0.3645E+00	0.3560E+00	-.5905E+00
0.2200E+02	0.8306E+00	0.6599E+00	0.6414E+00	-.5522E+00	0.4055E+00	0.4494E+00	-.6750E+00
0.2300E+02	0.8579E+00	0.6332E+00	0.6064E+00	-.6522E+00	0.4465E+00	0.5448E+00	-.7516E+00
0.2400E+02	0.8841E+00	0.6072E+00	0.5744E+00	-.7522E+00	0.4875E+00	0.6410E+00	-.8220E+00
0.2500E+02	0.9093E+00	0.5822E+00	0.5461E+00	-.8522E+00	0.5285E+00	0.7369E+00	-.8884E+00
0.2600E+02	0.9334E+00	0.5582E+00	0.5218E+00	-.9522E+00	0.5695E+00	0.8315E+00	-.9530E+00
0.2700E+02	0.9566E+00	0.5354E+00	0.5038E+00	-.1000E+01	0.6105E+00	0.9226E+00	-.9480E+00
0.2800E+02	0.9788E+00	0.5140E+00	0.5039E+00	-.1000E+01	0.6515E+00	0.9996E+00	-.9129E+00
0.2900E+02	0.1000E+01	0.4937E+00	0.5175E+00	-.1000E+01	0.6925E+00	0.1064E+01	-.9071E+00
0.3000E+02	0.1021E+01	0.4740E+00	0.5391E+00	-.1000E+01	0.7335E+00	0.1118E+01	-.9213E+00

* * * INTEGRATION SEGMENT 2

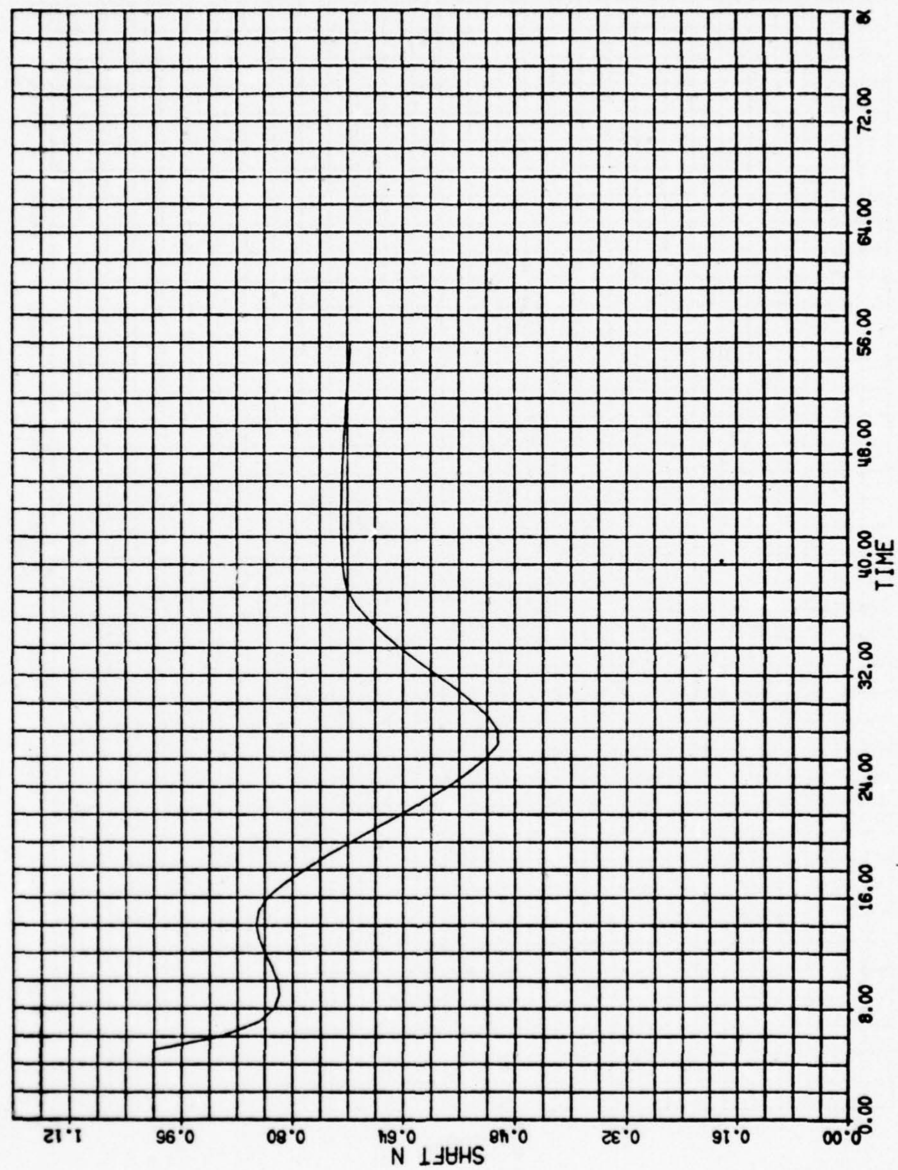
TIME	ADVANCE	VELOCITY	SHAFT N	PITCH/D	THROTTLE	ENGINE Q	PROP Q
0.3000E+02	0.1021E+01	0.4740E+00	0.5391E+00	-.1000E+01	0.7335E+00	0.1118E+01	-.9213E+00
0.3100E+02	0.1041E+01	0.4547E+00	0.5636E+00	-.1000E+01	0.7745E+00	0.1165E+01	-.9465E+00
0.3200E+02	0.1060E+01	0.4357E+00	0.5912E+00	-.1000E+01	0.8155E+00	0.1206E+01	-.9858E+00
0.3300E+02	0.1078E+01	0.4168E+00	0.6188E+00	-.1000E+01	0.8565E+00	0.1243E+01	-.1033E+01
0.3400E+02	0.1096E+01	0.3980E+00	0.6448E+00	-.1000E+01	0.8975E+00	0.1278E+01	-.1084E+01
0.3500E+02	0.1113E+01	0.3789E+00	0.6681E+00	-.1000E+01	0.9385E+00	0.1312E+01	-.1135E+01
0.3600E+02	0.1129E+01	0.3597E+00	0.6888E+00	-.1000E+01	0.9795E+00	0.1345E+01	-.1184E+01
0.3700E+02	0.1144E+01	0.3403E+00	0.7070E+00	-.1000E+01	0.1000E+01	0.1352E+01	-.1231E+01
0.3800E+02	0.1158E+01	0.3206E+00	0.7191E+00	-.1000E+01	0.1000E+01	0.1337E+01	-.1262E+01
0.3900E+02	0.1172E+01	0.3009E+00	0.7242E+00	-.1000E+01	0.1000E+01	0.1331E+01	-.1274E+01
0.4000E+02	0.1184E+01	0.2813E+00	0.7268E+00	-.1000E+01	0.1000E+01	0.1328E+01	-.1279E+01
0.4100E+02	0.1196E+01	0.2619E+00	0.7281E+00	-.1000E+01	0.1000E+01	0.1326E+01	-.1282E+01
0.4200E+02	0.1207E+01	0.2427E+00	0.7288E+00	-.1000E+01	0.1000E+01	0.1325E+01	-.1284E+01
0.4300E+02	0.1218E+01	0.2237E+00	0.7291E+00	-.1000E+01	0.1000E+01	0.1325E+01	-.1286E+01
0.4400E+02	0.1227E+01	0.2048E+00	0.7290E+00	-.1000E+01	0.1000E+01	0.1325E+01	-.1288E+01
0.4500E+02	0.1236E+01	0.1861E+00	0.7286E+00	-.1000E+01	0.1000E+01	0.1326E+01	-.1290E+01
0.4600E+02	0.1243E+01	0.1675E+00	0.7279E+00	-.1000E+01	0.1000E+01	0.1326E+01	-.1292E+01
0.4700E+02	0.1251E+01	0.1489E+00	0.7271E+00	-.1000E+01	0.1000E+01	0.1328E+01	-.1295E+01
0.4800E+02	0.1257E+01	0.1305E+00	0.7260E+00	-.1000E+01	0.1000E+01	0.1329E+01	-.1297E+01
0.4900E+02	0.1262E+01	0.1120E+00	0.7248E+00	-.1000E+01	0.1000E+01	0.1330E+01	-.1299E+01
0.5000E+02	0.1267E+01	0.9363E-01	0.7235E+00	-.1000E+01	0.1000E+01	0.1332E+01	-.1301E+01
0.5100E+02	0.1271E+01	0.7526E-01	0.7221E+00	-.1000E+01	0.1000E+01	0.1333E+01	-.1303E+01
0.5200E+02	0.1274E+01	0.5690E-01	0.7207E+00	-.1000E+01	0.1000E+01	0.1335E+01	-.1305E+01
0.5300E+02	0.1277E+01	0.3853E-01	0.7192E+00	-.1000E+01	0.1000E+01	0.1337E+01	-.1306E+01
0.5400E+02	0.1278E+01	0.2015E-01	0.7178E+00	-.1000E+01	0.1000E+01	0.1339E+01	-.1308E+01
0.5500E+02	0.1279E+01	0.1761E-02	0.7164E+00	-.1000E+01	0.1000E+01	0.1340E+01	-.1309E+01
0.5600E+02	0.1279E+01	-.1665E-01	0.7151E+00	-.1000E+01	0.1000E+01	0.1342E+01	-.1310E+01

*** VALUE OF VELOCITY HAS REACHED THE LIMITING VALUE OF -0.100000E-02
*** RUN TERMINATED.

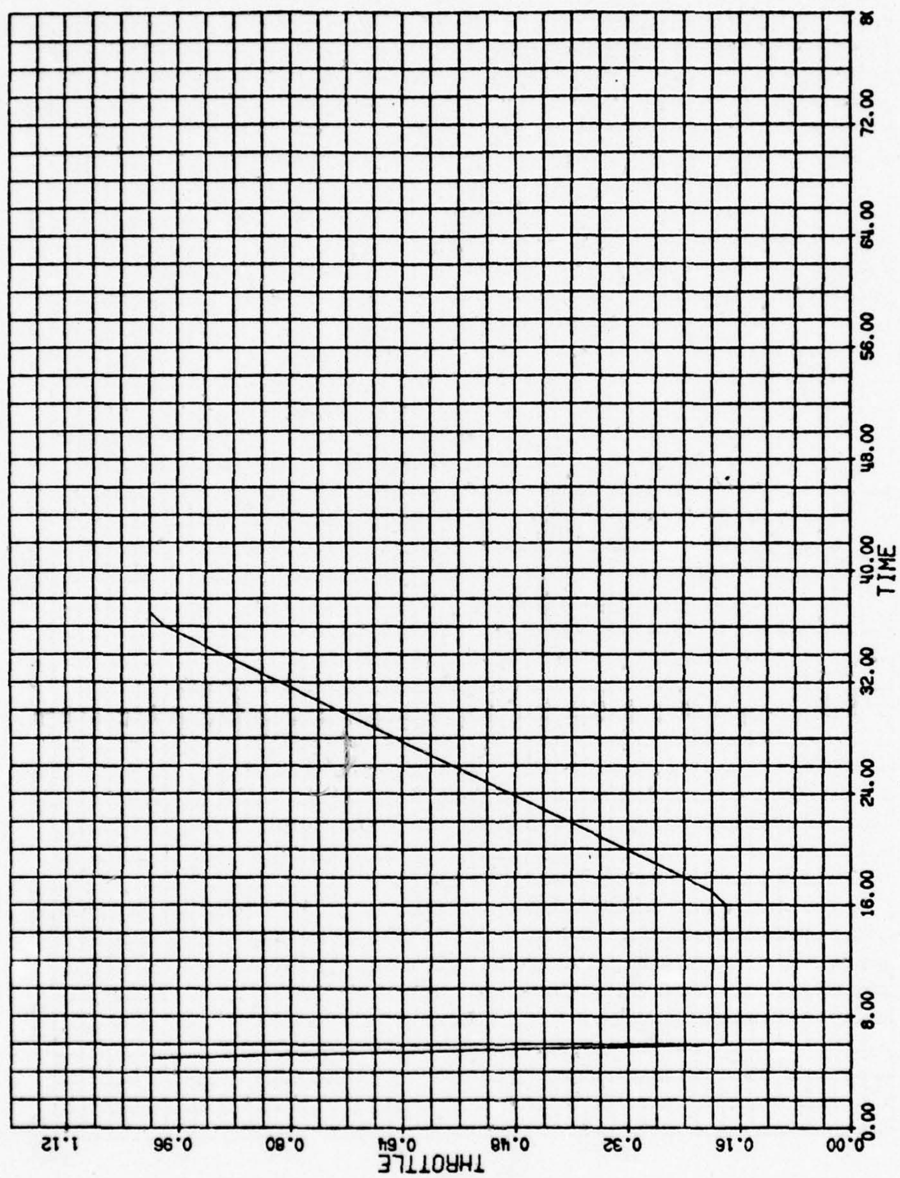
```
PDS: PLOT DESCRIPTION GENERATION BEGINS ***

INPUT VERIFICATION: STEADY STATE PITCH=  1.148 D(THROTTLE)/DT= 0.0820

CRASH STOP SIMULATION: IDLE TO FULL THROTTLE IN 10 SEC.

   * * * VARIABLES AND INITIAL VALUES:

         ADVANCE  = 0.0         VELOCITY = 0.100E+01  SHAFT N  = 0.100E+01

   * * * AUXILIARY VARIABLES:

         PITCH/D    THROTTLE  ENGINE Q  PROP Q

   * * * VARIABLES TO BE PRINTED:      ADVANCE   VELOCITY  SHAFT N

   * * * LIMITING VALUE OF VELOCITY IS -.1000E-02

   * * * INTEGRATION CONTROL PARAMETERS:

         SEGMENT  METHOD   TF           PRD          PLD          FIRSTP       EPS          AB          NCUTS

            1     EULER  0.3000E+02  0.1000E+01  0.1000E+01  0.1000E+00
            2     EULER  0.6000E+02  0.1000E+01  0.1000E+01  0.1000E+01



   * * * INTEGRATION SEGMENT 1

THE OUTPUT VECTOR IS:  ADVANCE    VELOCITY   SHAFT N    PITCH/D    THROTTLE
                       ENGINE Q   PROP Q

T= 0.0                 0.0         0.1000E+01  0.1000E+01  0.1148E+01  0.1000E+01
                       0.1000E+01  -.9697E+00

T= 0.1000E+01          0.4222E-01  0.1000E+01  0.1000E+01  0.1148E+01  0.1000E+01
                       0.1000E+01  -.9698E+00

T= 0.2000E+01          0.8445E-01  0.1000E+01  0.1000E+01  0.1148E+01  0.1000E+01
                       0.9999E+00  -.9699E+00

T= 0.3000E+01          0.1267E+00  0.1000E+01  0.1000E+01  0.1148E+01  0.1000E+01
                       0.9999E+00  -.9699E+00

T= 0.4000E+01          0.1689E+00  0.1000E+01  0.1000E+01  0.1148E+01  0.1000E+01
                       0.9999E+00  -.9699E+00

T= 0.5000E+01          0.2111E+00  0.1000E+01  0.1000E+01  0.1148E+01  0.1000E+01
                       0.9999E+00  -.9699E+00


T= 0.6000E+01          0.2533E+00  0.9962E+00  0.9014E+00  0.1048E+01  0.1800E+00
                       -.3521E-01  -.4491E+00

T= 0.7000E+01          0.2952E+00  0.9863E+00  0.8486E+00  0.9478E+00  0.1800E+00
                       -.2729E-01  -.1966E+00

T= 0.8000E+01          0.3366E+00  0.9734E+00  0.8254E+00  0.8478E+00  0.1800E+00
                       -.2382E-01  -.6132E-01

T= 0.9000E+01          0.3774E+00  0.9587E+00  0.8187E+00  0.7478E+00  0.1800E+00
                       -.2281E-01  0.1830E-01

T= 0.1000E+02          0.4176E+00  0.9427E+00  0.8216E+00  0.6478E+00  0.1800E+00
                       -.2324E-01  0.6486E-01

T= 0.1100E+02          0.4571E+00  0.9256E+00  0.8296E+00  0.5478E+00  0.1800E+00
                       -.2445E-01  0.8727E-01

T= 0.1200E+02          0.4958E+00  0.9073E+00  0.8394E+00  0.4478E+00  0.1800E+00
                       -.2592E-01  0.8865E-01

T= 0.1300E+02          0.5338E+00  0.8878E+00  0.8479E+00  0.3478E+00  0.1800E+00
                       -.2719E-01  0.6960E-01
```

```
T= 0.1400E+02          0.5709E+00  0.8670E+00  0.8519E+00  0.2478E+00  0.1800E+00
                       -.2779E-01  0.2992E-01

T= 0.1500E+02          0.6071E+00  0.8447E+00  0.8485E+00  0.1478E+00  0.1800E+00
                       -.2729E-01  -.2993E-01

T= 0.1600E+02          0.6423E+00  0.8209E+00  0.8351E+00  0.4778E-01  0.1800E+00
                       -.2528E-01  -.1076E+00

T= 0.1700E+02          0.6765E+00  0.7958E+00  0.8116E+00  -.5222E-01  0.2210E+00
                       0.5175E-01  -.1984E+00

T= 0.1800E+02          0.7096E+00  0.7694E+00  0.7924E+00  -.1522E+00  0.3030E+00
                       0.2016E+00  -.3020E+00

T= 0.1900E+02          0.7416E+00  0.7420E+00  0.7784E+00  -.2522E+00  0.3850E+00
                       0.3490E+00  -.4172E+00

T= 0.2000E+02          0.7724E+00  0.7138E+00  0.7678E+00  -.3522E+00  0.4670E+00
                       0.4929E+00  -.5419E+00

T= 0.2100E+02          0.8020E+00  0.6848E+00  0.7586E+00  -.4522E+00  0.5490E+00
                       0.6333E+00  -.6735E+00

T= 0.2200E+02          0.8303E+00  0.6555E+00  0.7496E+00  -.5522E+00  0.6310E+00
                       0.7706E+00  -.8090E+00

T= 0.2300E+02          0.8574E+00  0.6259E+00  0.7400E+00  -.6522E+00  0.7130E+00
                       0.9054E+00  -.9453E+00

T= 0.2400E+02          0.8833E+00  0.5966E+00  0.7295E+00  -.7522E+00  0.7950E+00
                       0.1038E+01  -.1080E+01

T= 0.2500E+02          0.9079E+00  0.5677E+00  0.7181E+00  -.8522E+00  0.8770E+00
                       0.1169E+01  -.1212E+01

T= 0.2600E+02          0.9314E+00  0.5396E+00  0.7061E+00  -.9522E+00  0.9590E+00
                       0.1297E+01  -.1339E+01

T= 0.2700E+02          0.9536E+00  0.5127E+00  0.6953E+00  -.1000E+01  0.1000E+01
                       0.1366E+01  -.1364E+01


T= 0.2800E+02          0.9748E+00  0.4872E+00  0.6929E+00  -.1000E+01  0.1000E+01
                       0.1369E+01  -.1321E+01

T= 0.2900E+02          0.9949E+00  0.4631E+00  0.6954E+00  -.1000E+01  0.1000E+01
                       0.1366E+01  -.1298E+01

T= 0.3000E+02          0.1014E+01  0.4399E+00  0.7000E+00  -.1000E+01  0.1000E+01
                       0.1360E+01  -.1285E+01


  * * * INTEGRATION SEGMENT 2

THE OUTPUT VECTOR IS:  ADVANCE   VELOCITY  SHAFT N   PITCH/D   THROTTLE
                       ENGINE Q  PROP Q

T= 0.3000E+02          0.1014E+01  0.4399E+00  0.7000E+00  -.1000E+01  0.1000E+01
                       0.1360E+01  -.1285E+01

T= 0.3100E+02          0.1033E+01  0.4172E+00  0.7051E+00  -.1000E+01  0.1000E+01
                       0.1354E+01  -.1278E+01

T= 0.3200E+02          0.1050E+01  0.3952E+00  0.7103E+00  -.1000E+01  0.1000E+01
                       0.1348E+01  -.1276E+01

                       .
                       .
                       .
```

```
T= 0.5000E+02           0.1223E+01  0.4880E-01  0.7200E+00  -.1000E+01  0.1000E+01
                        0.1336E+01  -.1306E+01

T= 0.5100E+02           0.1225E+01  0.3043E-01  0.7186E+00  -.1000E+01  0.1000E+01
                        0.1338E+01  -.1307E+01

T= 0.5200E+02           0.1226E+01  0.1204E-01  0.7172E+00  -.1000E+01  0.1000E+01
                        0.1339E+01  -.1308E+01

T= 0.5300E+02           0.1227E+01  -.6357E-02  0.7158E+00  -.1000E+01  0.1000E+01
                        0.1341E+01  -.1309E+01

*** VALUE OF VELOCITY HAS REACHED THE LIMITING VALUE OF   -0.100000E-02
*** RUN TERMINATED.

*** END OF FILE ENCOUNTERED ON 4.
#EXECUTION TERMINATED


 #RUN *CCQUEUE
 #EXECUTION BEGINS
  ENTER PLOT REQUEST:
  PLOT
    2 PLOTS; PLOTTING REQUIRES   450 SEC. AND    27 IN.;    $1.25
  OK?
  OK
  PLOT ASSIGNED RECEIPT # 510133.
  ENTER PLOT REQUEST:

 EOF


#EXECUTION TERMINATED
#
#DISPLAY COST
#COST =    $4.40, TERM,NORMAL,UNIV
#
#
#
#SIG
#SGWY 09:54:35-10:13:32 FRI JUN 10/77
#TERM,NORMAL,UNIV
#ELAPSED TIME        18.933 MIN.          $.48
#CPU TIME USED        6.636 SEC.         $2.18
#CPU STOR VMI         5.25  PAGE-MIN.     $.52
#WAIT STOR VMI       10.201 PAGE-HR.
#DRUM READS         406
#PLOT TIME            7.5   MIN.         $1.04
#PLOT PAPER           2.25  FEET          $.23
#APPROX. COST OF THIS RUN IS    $4.43
#DISK STORAGE       66      PAGE-HR.      $.01
```



CRASH STOP SIMULATION: IDLE TO FULL THROTTLE IN 20 SECONDS



CRASH STOP SIMULATION: IDLE TO FULL THROTTLE IN 20 SECONDS

APPENDIX B: User's Documentation for OPTSIM

UNIVERSITY OF MICHIGAN

DEPARTMENT OF NAVAL ARCHITECTURE AND MARINE ENGINEERING Rev. 1
5/27/77

IDENTIFICATION: OPTSIM: A group of subroutines

PROGRAMMER: Ass't Professor Michael G. Parsons and H.T. Cuong, Dept. of Naval Architecture and Marine Engineering, Univ. of Michigan, under ONR Contract N00014-76-C-0751.

PURPOSE: These subroutines are designed to be used in parallel with the OPTSYS program to simulate the response of stationary, linear optimal control and filter systems (developed using OPTSYS) to randomly generated measurement noise, initial condition errors, and specific process disturbances. These subroutines operate under the SHIPSIM continuous systems simulation program. The user inputs the system, estimator, control gain, and filter gain matrices and the standard deviations of the measurement noise. Initial condition errors can be input through SHIPSIM. A subroutine DISTRB is provided by the user to generate the specific process disturbance of interest. A second subroutine ADERIV can also be provided by the user if additional variables must be integrated to generate the desired output.

METHOD:

References:

1. University of Michigan, Department of Naval Architecture and Marine Engineering, OPTSYS Program, Rev. 2, 5/27/77

2. University of Michigan, Department of Naval Architecture and Marine Engineering, SHIPSIM Program, Rev. 2, 5/27/77

3. "IBM System/360 and System/370 FORTRAN IV Language," IBM Manual GC28-6515-10.

This set of double precision subroutines provides a general way to simulate the response of a stationary, linear optimal controller and Kalman-Bucy filter system to randomly generated measurement noise, initial condition errors, and specific process disturbances. These systems are designed for stochastic process noise models such as white noise or the output of various shaping filters. The OPTSYS program[1] can be used to design the controller and filter and evaluate the RMS response of the controlled system when subjected to the design measurement noise and process disturbances. It is often desirable to also simulate such a system to establish the response of the controlled system to specific, realistic initial condition errors and process disturbances.

For the stationary, linear system disturbed by Gauss-Markov noise,

$$\dot{\underline{x}} = F\underline{x} + G\underline{u} + \Gamma\underline{w} \tag{1}$$

$$\underline{z} = H\underline{x} + \underline{v} \quad , \tag{2}$$

where $\underline{x}$ may be augmented to include the process noise generated by shaping filters as additional states, the OPTSYS program provides the optimal control,

$$\underline{u} = C\hat{\underline{x}} \quad , \tag{3}$$

which uses the estimate of the state produced by the Kalman-Bucy filter given by,

$$\dot{\hat{\underline{x}}} = F\hat{\underline{x}} + G\underline{u} + K(\underline{z}-H\hat{\underline{x}}) \quad . \tag{4}$$

OPTSYS can also be used to establish the RMS response of this controlled system when subjected to the design white noise sources $\underline{w}$ and $\underline{v}$. It may also be desirable to establish the controlled system response to initial condition errors $\underline{x}(t_o)$ or $(\underline{x}(t_o)-\hat{\underline{x}}(t_o))$ and specific process disturbances $\underline{w}$ while experiencing the measurement white noise $\underline{v}$.

The OPTSIM subroutines are designed to perform this simulation under the control of the SHIPSIM continuous systems simulation program[2]. OPTSIM constitutes the INPUT and DERIV subroutines required by SHIPSIM so these do not need to be provided by the user. To the extent possible the input formats are the same as those used by OPTSYS so the same data sets can be utilized. For the most general case where shaping filters are used to model the process disturbances, the estimator estimates both the states and the disturbances (augmented states). In this situation, the filter is of higher order than the system and the formulation can be as follows:

$$\dot{\underline{x}} = F_s\underline{x} + G_sC\,\hat{\underline{x}} + \Gamma\underline{w} \quad , \tag{5}$$

$$\dot{\hat{\underline{x}}} = KH_s\underline{x} + (F_e + G_eC - KH_e)\hat{\underline{x}} + K\underline{v} \quad , \tag{6}$$

where:

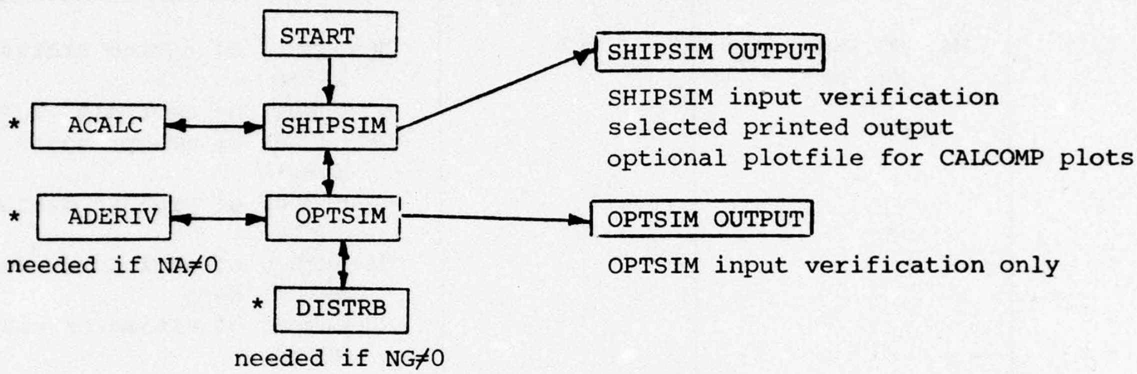
- $\underline{x}$ = system state vector <u>without</u> augmented states (NSx1)
- $\hat{\underline{x}}$ = estimator state vector <u>with</u> augmented states (NEx1)
- $\underline{w}$ = process disturbance vector (NGx1)
- $\underline{v}$ = measurement noise vector (NOBx1) with standard deviations $\underline{\sigma}$(NOBx1)
- F_s = system open-loop dynamics matrix (NSxNS)
- F_e = estimator open-loop dynamics matrix (NExNE)
- G_s = system control distribution matrix (NSxNC)
- G_e = estimator control distribution matrix (NExNC)
- Γ = system disturbance distribution matrix (NSxNG)
- H_s = system measurement scaling matrix (NOBxNS)
- H_e = estimator measurement scaling matrix (NOBxNE)
- C = feedback control gains (NCxNE)
- K = Kalman-Bucy filter gains (NExNOB)

If no shaping filters are used NS=NE and $F_s=F_e$, $G_s=G_e$ and $H_s=H_e$.

If it is desired to integrate variables in addition to $\underline{x}$ and $\hat{\underline{x}}$, these derivatives can be included by adding an optional subroutine ADERIV which calculates,

$$\dot{\underline{y}} = \underline{f}(t,\underline{x},\hat{\underline{x}},\underline{y}) \tag{7}$$

where $\underline{y}$ is the vector of additional variables (NAx1). If it is desired to perform calculations at any print or plot point to produce output in addition to $t,\underline{x},\hat{\underline{x}}$, and $\underline{y}$, this can be accomplished by adding an optional, user provided subroutine ACALC as defined by SHIPSIM.[2] If NG≠0, the user must provide a subroutine DISTRB which will provide the process disturbance $\underline{w}$ as a function time. The general structure of the entire program, OPTSIM running under control of SHIPSIM, is as follows:



*optional user supplied subroutines; with dummy subroutines available

The input is structured so that multiple runs can be made using a completely new data set , by changing only SHIPSIM integration control parameters, or by changing only specific terms within any of the matrices in equations (5) or (6).

USER'S INSTRUCTIONS: The input data sets consist of simulation specification and control input required by SHIPSIM[2] (which will not be detailed here) on I/O device 4 and the OPTSIM input described below in I/O device 5. Multiple runs can be made by following these with additional data for SHIPSIM and OPTSIM. Data sets for two runs are read in the following order:

- OPTSIM data for first run
- SHIPSIM data for first run
- input control variable NEXT for SHIPSIM
- new OPTSIM data or changes to first run OPTSIM data for second run
- new SHIPSIM data or changes to first run SHIPSIM data for second run

The OPTSIM data for an initial run or a completely new run (SWITCH=1) should consist of the following in the specified sequence including only those required:

Record Type	Input	Format	Comments
1.	SWITCH (must equal 1 for an initial run)	I3	SWITCH=1 if a completely new set of data will be given SWITCH=2 if only specific changes will be given using NAMELIST
2.	NS, NC, NOB, NG, NA, NE	6I3	NS=number of system states, $NS \leq 10$ NC=number of controls, $NC \leq 8$ NOB=number of measurements, $NOB \leq 10$ NG=number of process disturbances, $NG \leq 10$ NA=number of additional variables, $NA \leq 5$ NE=number of estimator states, $NE \leq 10$
3.	FS	6E12.5 repeated as required	F_s matrix input by rows beginning with a <u>new</u> record for each row
4.*	FE (only if NE≠NS)	6E12.5 repeated as required	F_e matrix input by rows beginning with a <u>new</u> record for each row
5.	GS	6E12.5 repeated as required	G_s input by rows <u>without</u> a new record for each row
6.*	GE (only if NE≠NS)	6E12.5 repeated as required	G_e input by rows <u>without</u> a new record for each row
7.	C	6E12.5 repeated as required	C input by rows <u>without</u> a new record for each row
8.*	GAMMA (only if NG≠0)	6E12.5 repeated as required	Γ input by rows <u>without</u> a new record for each row

Record Type	Input	Format	Comments
9.	HS	6E12.5 repeated as required	H_s input by rows without a new record for each row
10.*	HE (only if NE≠NS)	6E12.5 repeated as required	H_e input by rows without a new record for each row
11.	K	6E12.5 repeated as required	K input by rows without a new record for each row
12.	SIGMA	6E12.5 repeated as required	σ'; input a zero vector if no measurement noise is desired See discussion beginning at the bottom page B-6 concerning the specification of σ'.

*included only if required by input on Card 2.

The form of OPTSIM data for subsequent runs is controlled by the variable SWITCH. If SHIPSIM data is changed but the existing (previous run) OPTSIM data is to be used again, SHIPSIM variable NEXT should equal 1 or 3 and no OPTSIM data is needed. If SWITCH is set equal to 1 on record type 1, a completely new data set must be supplied as described above. If SWITCH is set equal to 2 on record type 1, any specific elements of F_s, F_e, G_s, G_e, C, Γ, H_s, H_e, K, and σ can be changed in the existing OPTSIM data using the format-free NAMELIST input.[3] This input would consist of a record type 1 followed by a record or records as follows:

```
column 123456789
        &LIST1 F(1,3)=0.0,C=1.0,5*0.0,...&END
```

In this example, element F(1,3) is zeroed and the control gains matrix C(2x3) is changed to 1.0 in the first element with the remaining elements zero. The input can be continued to additional records provided each continued record ends with a comma and each continuing record begins with a new variable name. If all data alterations can be made on a single record, the &LIST1 and the &END can be omitted and the first variable name can begin in column 1. A user should review the use of NAMELIST input in the IBM FORTRAN IV language manual.[3]

Subroutine ADERIV. IF NA=0, the user can call a dummy subroutine as part of the MTS RUN command as described below and there is no need to write ADERIV. If NA≠0, the user must write ADERIV and load the compiled subroutine into a file to be referenced on the MTS RUN command. This subroutine should

appear as follows:

```
      SUBROUTINE ADERIV(TIME,X,YDOT,NA,NEQ)
      IMPLICIT REAL*8(A-H,O-$)
      DIMENSION X(NEQ),YDOT(NA)
          .
          .
          .
        code which calculates the vector of NA additional derivatives
         YDOT(NA)
          .
          .
          .
      RETURN
      END
```

Here TIME is the independent variable, X(NEQ) is the full vector of variables being integrated (NEQ=number of equations) in which $\underline{x}$ occurs first, $\underline{\hat{x}}$ occurs next, and the additional variables $\underline{y}$ appear last. Thus NEQ=NS+NE+NA. This information can be used if needed to calculate the YDOT at any point in time.

Subroutine DISTRB. If NG=0, the user can call a dummy subroutine as part of the MTS RUN command as described below and there is no need to write DISTRB. If NG≠0, the user must write DISTRB and load the compiled subroutine into a file to be referenced on the MTS RUN command. This subroutine should appear as follows:

```
      SUBROUTINE DISTRB (TIME,X,W,NG,NEQ)
      IMPLICIT REAL*8(A-H,O-$)
      DIMENSION W(NG),X(NEQ)
          .
          .
          .
        code which calculates the process disturbance vector W(NG)
          .
          .
          .
      RETURN
      END
```

Here TIME and X(NEQ) are as defined above.

The user will also have to prepare SHIPSIM input as described in the User's Documentation for SHIPSIM.[2] This will include the selection of integration method and specification of integration control parameters. For a system subjected to white noise disturbances or measurement noise, the integration method choice, step size choice, and specification of measurement noise standard deviation (σ in OPTSIM input) must be done with care. First, only fixed step-size Euler or rectangular integration should be specified. This has the effect of approximating the gauss-markov continuous process by a discrete gauss-markov sequence. (See Bryson and Ho, Applied Optimal Control, Blaisdell, 1969, pp. 342-344 and pp. 364-366). In order for this approximation to preserve the correct system response covariance, the white noise for measurement i must have a standard deviation given by,

$$\sigma_i' = \left[\frac{R_{ii}}{\Delta t}\right]^{1/2}$$

where R_{ii} is the corresponding power spectral density used for measurement i in the OPTSYS design and Δt is the simulation integration step-size specified in the SHIPSIM input. Since σ_i' depends on Δt, the variable step-size Kutta-Merson integration available in SHIPSIM must not be used.

MTS RUN INFORMATION: The object code for OPTSIM and the dummy subroutines ADERIV and DISTRB are in files under account SGTA (subject to change; check with Prof. Michael G. Parsons). The RUN is actually made using SHIPSIM so reference should also be made to its description and user's instructions.[2] It is also necessary to utilize two subroutines out of the IBM Scientific Subroutine Package in calculating the random measurement noise. A run using the three dummy subroutines would appear as follows:

```
$RUN SGTA:SHIPSIM.O+SGTA:ACALC.D+SGTA:OPTSIM.O+SGTA:ADERIV.D+SGTA:DISTRB.D
+NAAS:SSP+*PLOTSYS 4=(SHIPSIM input data file) 5=(OPTSIM input data file)
9=(output file for input to *PLOTSYS)
```

Appropriate user supplied object code files should be referenced if the optional subroutines ACALC (see SHIPSIM description[2]), ADERIV, and/or DISTRB are utilized. If device 9 is not used it can be set equal to *DUMMY* or a temporary file.

A complete example will not be included here; example runs can be made available upon request. A sample OPTSIM input data file and the corresponding input verification output produced by OPTSIM for a case where NS=NE follows. Sample output from SHIPSIM is included in the SHIPSIM User's Documentation.[2]

OPTSIM input file:

	1
	5 1 3 2 1 5
F_s	0.0E0 1.0E0 0.0E0 0.0E0 0.0E0
	0.0 -0.17657E+01 0.57359E+01 0.0 -0.88074E+00
	0.0 0.17199E+00-0.52766E+00 0.0 -0.15607E+00
	1.0E0 0.0E0 -1.0E0 0.0E0 0.0E0
	0.0E0 0.0E0 0.0E0 0.0E0 -0.439E0
G_s	0.0E0 0.0E0 0.0E0 0.0E0 0.439E0
C	7.74621E+00 4.62370E+00 1.70009E+01 2.42523E+00-4.71320E+00
Γ	0.0 0.0 0.47768E+03-0.50043E+01 0.21141E+02-0.28233E+02
	0.0 0.0 0.0 0.0
H_s	1.0E0 0.0E0 0.0E0 0.0E0 0.0E0 0.0E0
	1.0E0 0.0E0 0.0E0 0.0E0 0.0E0 0.0E0
	0.0E0 1.0E0 0.0E0
K	2.18897E-01 1.00973E+00 1.35941E-03 4.85688E-02 6.18658E+02-4.43003E-03
	6.71746E-02 4.56445E+01-4.76066E-01 1.32662E-01-8.98771E+00 9.56480E-01
	0.0 0.0 0.0
σ	3.4904E-3 7.6544E-4 3.4480E-2

OPTSIM input verification output:

```
OPTSIM OPTIMAL STOCHASTIC CONTROLLER SIMULATION PROGRAM

INPUT VERIFICATION    NEQ = 11

ORDER OF SYSTEM =  5

NUMBER OF CONTROLS =  1

NUMBER OF OBSERVATIONS =  3

NUMBER OF PROCESS NOISE SOURCES =  2

NUMBER OF AUXILIARY STATES =  1

ORDER OF ESTIMATOR =  5

SYSTEM OPEN LOOP DYNAMICS MATRIX....FS(NS,NS)

   0.0            0.10000E+01   0.0           0.0           0.0
   0.0           -0.17657E+01   0.57359E+01   0.0          -0.88074E+00
   0.0            0.17199E+00  -0.52766E+00   0.0          -0.15607E+00
   0.10000E+01    0.0          -0.10000E+01   0.0           0.0
   0.0            0.0           0.0           0.0          -0.43900E+00

SYSTEM CONTROL DISTRIBUTION MATRIX....GS(NS,NC)

   0.0
   0.0
   0.0
   0.0
   0.43900E+00

FEEDBACK CONTROL GAINS....C(NC,NS)

   0.77462E+01    0.46237E+01   0.17001E+02   0.24252E+01  -0.47132E+01

SYSTEM DISTURBANCE DISTRIBUTION MATRIX....GAMMA(NS,NG)

   0.0            0.0
   0.47768E+03   -0.50043E+01
   0.21141E+02   -0.28233E+02
   0.0            0.0
   0.0            0.0

SYSTEM MEASUREMENT SCALING MATRIX....HS(NOB,NS)

   0.10000E+01    0.0           0.0           0.0           0.0
   0.0            0.10000E+01   0.0           0.0           0.0
   0.0            0.0           0.0           0.10000E+01   0.0

KALMAN-BUCY FILTER GAINS....K(NS,NOB)

   0.21890E+00    0.10097E+01   0.13594E-02
   0.48569E-01    0.61866E+03  -0.44300E-02
   0.67175E-01    0.45645E+02  -0.47607E+00
   0.13266E+00   -0.89877E+01   0.95648E+00
   0.0            0.0           0.0

MEASUREMENT NOISE STANDARD DEVIATIONS....SIGMA(NOB)

   0.34904E-02
   0.76544E-03
   0.34480E-01
```

Appendix C: User's Documentation for OPTSYS

UNIVERSITY OF MICHIGAN

DEPARTMENT OF NAVAL ARCHITECTURE AND MARINE ENGINEERING Rev. 2
5/27/77

IDENTIFICATION: OPTSYS Program

PROGRAMMER: original version by W. Earl Hall, Jr., Dept. of Aeronautics and Astronautics, Stanford Univ., 1971, (now Systems Control, Inc., Palo Alto, CA.); adapted to MTS by Ass't Professor Michael G. Parsons, Dept. of Naval Architecture and Marine Engineering, Univ. of Michigan, June, 1976.

PURPOSE: The program provides fast and efficient solutions to the steady-state, linear optimal control and filter problems using the usual quadratic penalty function on the state and control variables. With user inputs of penalty function, constant coefficient state and measurement equations, and noise spectra, the various program options will provide optimal feedback gains, Kalman-Bucy filter gains, RMS state and control response, response to a constant disturbance, and an evaluation of controllability, disturbability and observability via modal decomposition of the state and measurement equations. It is also possible to evaluate RMS state and control response when feedback gains and filter gains are provided externally.

METHOD:

References:

1. Bryson, A.E. Jr., and Ho, Y.C., Applied Optimal Control, Blaisdell Publ., 1969.
2. Bryson, A.E. Jr., "Control Theory for Random Systems," Proceedings of the Thirteenth International Congress of Theoretical and Applied Mechanics, Moscow, August 1972.
3. MacFarlane, A.G.J., "An Eigenvector Solution of the Optimal Linear Regulator Problem," Journal of Electronics and Control, Vol. 14, No. 3, May, 1963, pp. 643-654.
4. Potter, J.E., "Matrix Quadratic Solutions," SIAM Journal of Applied Mathematics, Vol. 14, No. 3, May, 1966, pp. 496-501.
5. Bryson, A.E., Jr., and Hall, W.E. Jr., "Optimal Control and Filter Synthesis by Eigenvector Decomposition," Stanford Univ. Guidance and Control Lab. SUDAAR #436, Nov., 1971.

This program treats the problem of designing optimal controllers for stationary, linear systems disturbed by Gauss-Markov noise. The system must be represented by the model,

$$\dot{\underline{x}} = F\underline{x} + G\underline{u} + \Gamma\underline{w} , \tag{1}$$

$$\underline{z} = H\underline{x} + \underline{v} , \tag{2}$$

where:

$\underline{x}$ = state vector (nxl),
$\underline{u}$ = control vector (mxl),
$\underline{w}$ = white disturbance noise (qxl) with zero mean and power spectral density matrix Q (qxq); or separately a constant disturbance $\underline{w}_c$,
$\underline{z}$ = measurement vector (pxl),
$\underline{v}$ = white measurement noise (pxl) with zero mean and power spectral density matrix R(pxp),
F = open-loop dynamics matrix (nxn),
G = control distribution matrix (nxm),
Γ = state disturbance distribution matrix (nxq),
H = measurement scaling matrix (pxn).

An optimal controller can be defined as the one which minimizes the expected value of the quadratic penalty function,

$$J = \frac{1}{2}\underline{x}^T A\underline{x} + \frac{1}{2}\underline{u}^T B\underline{u}, \qquad (3)$$

where:

A = positive semi-definite state weighting matrix (nxn)
B = positive definite control weighting matrix (mxm).

This program utilizes the certainty-equivalence principle or separation principle which states that the optimal feedback in the ensemble average sense for the above system is the optimal deterministic controller preceded by the optimal estimator (or filter) of the states. The program can in sequence or individually design the optimal deterministic controller, design the Kalman-Bucy filter, and evaluate the RMS state and control response.

State-feedback Controller: The calculus of variations can be used to show[1,2] that the state-feedback controller which minimizes eqn. (3) is given by,

$$\underline{u} = C\underline{x} = -B^{-1}G^T S_\infty \underline{x}, \qquad (4)$$

where S_∞ is the steady-state solution of the backward matrix Riccati equation,

$$\dot{S} = -SF - F^T S + SGB^{-1}G^T S - A, \quad S(t_f) = 0. \qquad (5)$$

Most methods for solving for S_∞ (integrating eqn, (5) to steady-state or solving simultaneous quadratic equations) are expensive and subject to numerical difficulties for large n. This program utilizes a technique called eigenvector decomposition which was first proposed by MacFarlane[3] and Potter[4] and which is both fast and well-behaved. Earl Hall[5] used the QR algorithm for finding eigenvalues and eigenvectors to develop a practical and efficient means of utilizing the MacFarlane-Potter method. Hall's work was extended by other students at Stanford Univ. to produce this program.

The eigenvector decomposition method is based on finding the eigenvectors of the Euler-Lagrange equations (2n) for minimizing eqn. (3) subject to eqn. (1):

$$\begin{bmatrix} \dot{\underline{x}} \\ \dot{\underline{\lambda}} \end{bmatrix} = \begin{bmatrix} F & -GB^{-1}G^T \\ -A & -F^T \end{bmatrix} \begin{bmatrix} \underline{x} \\ \underline{\lambda} \end{bmatrix}. \tag{6}$$

The eigenvalues of this system are in pairs which are symmetric about the imaginary axis in the complex plane. If these eigenvalues are grouped into those with positive real parts and those with negative real parts, the associated eigenvectors can likewise be grouped as follows.

$$\tilde{\lambda} = \left[\begin{array}{c|c} X_+ & X_- \\ \hline \Lambda_+ & \Lambda_- \end{array}\right], \tag{7}$$

where the eigenvectors with minus subscript are associated with the eigenvalues with negative real parts, etc. The steady-state solution to eqn. (5) is given by,

$$S_\infty = \Lambda_- (X_-)^{-1}, \tag{8}$$

the eigenvalues of the controlled system (F-GC) are the eigenvalues of eqn. (6) with negative real parts, and the eigenvectors of the controlled system are the columns of X_-. Similarly, if eqn. (5) were a forward matrix Riccati equation the steady-state solution would be given by $-X_+ (\Lambda_+)^{-1}$ as will be utilized below in the filter problem.

<u>Kalman-Bucy Filter</u>: The calculus of variations can be used to show [1,2] that the maximum likelihood filter for estimating the state of a system disturbed by white noise from measurements which contain white noise is given by,

$$\dot{\hat{\underline{x}}} = F\hat{\underline{x}} + G\underline{u} + K(\underline{z}-H\hat{\underline{x}}), \tag{9}$$

where the filter gain matrix K is given by,

$$K = P_\infty H^T R^{-1}. \tag{10}$$

The matrix P_∞ is the steady-state solution of the forward matrix Riccati equation,

$$\dot{P} = FP + PF^T - PH^T R^{-1} HP + \Gamma Q \Gamma^T, \qquad P(t_o) = X(t_o). \tag{11}$$

Here P (nxn) is the covariance matrix of the error of the state estimate and X (nxn) is the covariance matrix of the state.

The steady-state solution of eqn. (11) may be found by using the eigenvector decomposition of the Euler-Lagrange equations (2n) for the filter problem [1,2]; i.e.,

$$\begin{bmatrix} \dot{\underline{x}} \\ \dot{\underline{\lambda}} \end{bmatrix} = \begin{bmatrix} F & -\Gamma Q \Gamma^T \\ -H^T R^{-1} H & -F^T \end{bmatrix} \begin{bmatrix} \underline{x} \\ \underline{\lambda} \end{bmatrix} + \begin{bmatrix} 0 \\ H^T R^{-1} Z \end{bmatrix} . \tag{12}$$

Since eqn. (11) is a forward matrix Riccati equation, the steady-state solution is given by,

$$P_\infty = -X_+ (\Lambda_+)^{-1} \tag{13}$$

where X_+ and Λ_+ are partitions of the eigenvectors associated with the eigenvalues of eqn. (12) having positive real parts. The eigenvalues of the estimate errors $\tilde{\underline{x}} = \hat{\underline{x}} - \underline{x}$ are the eigenvalues of eqn. (12) with negative real parts and the eigenvectors of the estimate errors are the columns of $(\Lambda_+)^{-1}$.

<u>RMS State and Control Response</u>: If perfect measurements are available (no filter), the stationary statistical response of the states is determined by the linear matrix equation[1] ,

$$(F+GC)X + X(F+GC)^T + \Gamma Q \Gamma^T = 0 \tag{14}$$

where again X is the covariance matrix of the states. The RMS response of each state is the square root of the associated diagonal element of X. The covariance matrix for the control will be,

$$E(\underline{u}\underline{u}^T) = CXC^T, \tag{15}$$

with the RMS control the square root of the associated diagonal element of this matrix.

If filtered estimates of the state are used, the covariance matrix for the state is given by,[1]

$$X = \hat{X} + P_\infty , \tag{16}$$

where $\hat{X}$ is the covariance matrix for the state estimate given by the linear matrix equation,

$$(F+GC)\hat{X} + \hat{X}(F+GC)^T + KRK^T = 0 \tag{17}$$

In this case, covariance matrix for the control is given by,

$$E(\underline{u}\underline{u}^T) = C\hat{X}C^T . \tag{18}$$

<u>Steady-state Response</u>: The stochastic control and RMS response are with respect to zero mean disturbances. The program will also separately calculate state and control response to a constant disturbance, if desired, by using,

$$\underline{x}_{ss} = -(F+GC)^{-1} \Gamma \underline{w}_c , \tag{19}$$

and,

$$\underline{u}_{ss} = C\underline{x}_{ss} \cdot \tag{20}$$

The calculated RMS response would then be with respect to or additive to these steady-state values.

<u>Controllability, Disturbability, and Observability</u>: This program can also be used to determine the relative effectiveness of selected control inputs, the relative coupling of state disturbances to the dynamic plant modes, and the relative detectability of the dynamic plant modes from selected measurements. The state and measurement equations can be displayed in modal (normal-coordinate or Jordon canonical) form; i.e.,

$$\dot{\underline{\xi}} = F'\underline{\xi} + G'\underline{u} + \Gamma'\underline{w} \tag{21}$$

$$\underline{z} = H'\underline{\xi} + \underline{v} \tag{22}$$

where:

$\underline{x} = T\underline{\xi}$,
T = matrix of right eigenvectors of F; columns are eigenvectors of $F\underline{t}_i = \underline{t}_i \lambda_i$,
$F' = T^{-1}FT$ is the diagonal eigenvalue matrix for distinct eigenvalues,
T^{-1} = matrix of left eigenvectors of F; rows are eigenvectors of $\underline{t}_i F = \underline{t}_i \lambda_i$,
$G' = T^{-1}G$,
$\Gamma' = T^{-1}\Gamma$,
$H' = HT$.

The degree of controllability (by control inputs) and disturbability (by state disturbances) are shown by displaying the degree of orthogonality between the control distribution vector(s) and disturbance distribution vector(s), respectively, and the left eigenvector of each open-loop mode. Thus for controllability of mode i with control j the program displays,

$$\cos \Theta_{ij} = \frac{|\underline{t}_{li}\underline{g}_j|}{|\underline{t}_{li}|\ |\underline{g}_j|}, \tag{23}$$

where $\underline{t}_{li}$ is the left eigenvector associated with mode i and $\underline{g}_j$ is the vector (column) of G associated with control j. A result 1.0 indicates maximum coupling or controllability and a result of 0.0 indicates complete uncontrollability. Similarly, the degree of disturbability is shown by displaying,

$$\cos\phi_{ij} = \frac{|\underline{t}_{li}\ \gamma_j|}{|\underline{t}_{li}|\,|\gamma_j|}, \tag{24}$$

where $\underline{\gamma}_j$ is the vector (column) of Γ associated with state disturbance j.

The degree of observability is shown by displaying the degree of orthogonality between the right eigenvector of each open-loop mode and the measurement distribution vector(s). Thus for observability of mode i with measurement j the program displays,

$$\cos\beta_{ij} = \frac{|\underline{h}_j \underline{t}_{ri}|}{|\underline{h}_j| |\underline{t}_{ri}|}, \tag{25}$$

where $\underline{t}_{ri}$ is the right eigenvector associated with mode i and $\underline{h}_j$ is the vector (row) of H associated with measurement j. A result 1.0 indicates maximum observability and a result 0.0 indicates that mode i is not observable with measurement j. If mode i is not observable from any measurement, the system is not observable.

For complex eigenvalues of F, the eigenvectors in T and T^{-1} are also complex. The elements of G', Γ', and H' will also be complex. The eigenvector matrices and G', Γ', and H' are then presented in a real form; i.e.,

$$T = [\underline{t}_{1r}, \underline{t}_{1i}, \underline{t}_{2r}, \underline{t}_{2i}, \ldots],$$

where $\underline{t}_{1r}$ and $\underline{t}_{1i}$ are, for example, the real and imaginary parts, respectively, of the first conjugate pair of right eigenvectors. Likewise, for complex eigenvalues the degree of controllability, disturbability, and observability are presented for the real and imaginary parts of the associated left (or right) eigenvector conjugate pair. Thus, $\cos\Theta_{1j}$ and $\cos\Theta_{2j}$ will be the results for the orthogonality of the real and imaginary part, respectively, of first pair of conjugate eigenvectors and the jth control distribution vector.

<u>PROGRAM OPTIONS</u>: This program is very flexible with a number of options which control output and needed input. These will be presented prior to defining the input details.

IOL = 1, if open-loop eigensystem is to be calculated and printed out;
  = 0, if not;

IQ = 1, if RMS state and control values are to be calculated and printed out;
  = 0, if not;

INQ = 0, if A,B,Q, and R will be input as diagonals only, off-diagonal elements assumed zero;
  = 1, if A and Q will be input as full matrices; B and R input as diagonals only, off-diagonal elements assumed zero;

IR = 0, if optimal feedback controller and filter are to be determined with C and K output;
  = 1, if C is to be provided as input; optimal filter to be determined and K output;
  = 2, if K is to be provided as input; optimal feedback controller to to be determined and C output;
  = 3, if both C and K are to be provided as input;

ISS = 1, if steady-state values for states and control for a steady disturbance $\underline{w}_c$ are to be determined and output; = 0, if not;

IM = 1, if modal equations (G', Γ', and H'), controllability, disturbability, and observability are to be determined and results output (IOL = 1, required); = 0, if not.

IP = 1, program calculated C and K matrices are to be output to I/O device 7 using format 6E12.5 repeated as required but beginning C and K with a new record; = 0, if not.

These options will allow the design and evaluation of an optimal controller and filter (IQ = 1, IR = 0); study of controllability, disturbability, and observability of a system (IOL = 1, IM = 1); evaluation of controller and filter performance when system characteristics change (IQ = 1, IR = 3, with new state equations F and G); etc. When control gains are provided as input the program checks the closed loop system for stability. Likewise, when filter gains are provided as input the filter is checked for stability. In each case, only that input necessary for the selected options must be provided.

USER'S INSTRUCTIONS: Input records should consist of the following in the specified sequence including only those required.

Record Type Number	Input	Format	Comments
1.	"title"	18A4	Any 72 character title for the output.
2.	IOL, IQ, INQ, IR, ISS, IM, IP	7I2	Program option control integers as defined above.
3.	NS, NC, NOB, NG	4I3	NS = number of states, $n \leq 25$ NC = number of controls, $m \leq 8$ NOB = number of measurements, $p \leq 25$ NG = number of state disturbances, $q \leq 25$
4.	F	6E12.5 repeated as required	F matrix input by rows beginning with a new record for each row.
5.*	A	6E12.5 repeated as required	A input as a diagonal only or input as an entire matrix as set by INQ. Matrix input by rows without a new record for each row.
6.*	G	6E12.5 repeated as required	G input by rows without a new record for each row.

Record Type Number	Input	Format	Comments
7.*	B	6E12.5 repeated as required	B input as diagonal only
8.*	C	6E12.5 repeated as required	C input by rows <u>without</u> a new record for each row if IR = 1 or 3.
9.*	Γ	6E12.5 repeated as required	Γ input by rows <u>without</u> a new record for each row.
10.*	Q	6E12.5 repeated as required	Refer to A above. This power spectral density defines $\underline{w}$.
11.*	H	6E12.5 repeated as required	H input by rows <u>without</u> a new record for each row.
12.*	R	6E12.5 repeated as required	R input as diagonal only. This power spectral density defines $\underline{v}$.
13.*	K	6E12.5 repeated as required	K input by rows <u>without</u> a new record for each row if IR = 2 or 3.
14.*	$\underline{w}_c$	6E12.5 repeated as required	steady disturbance vector only if ISS = 1.
15.	"job control"	A2	␢* if another case follows /* if no case follows

*included only if required by input on records 2 or 3.

To clarify which of the input record types 5 through 14 must be provided note that the input sequence is as follows:

$$F \mid A\ G\ B\ C \mid \Gamma\ Q \mid H\ R\ K \mid \underline{w}_c$$

Portions of this sequence should be deleted as follows:

```
if NC = 0, delete        A,G,B,C,
if NOB = 0, delete       H,R,K
if NG = 0, delete        Γ,Q,H,R,K
if ISS = 0, delete       w_c
if IR = 0, delete        C,K
      = 1, delete        A,B,K
      = 2, delete        C
      = 3, delete        A,B
```

<u>MTS RUN INFORMATION:</u> The object code for the OPTSYS program is on file under account SGTA (subject to change; check with Ass't Prof. Michael G. Parsons). The program can therefore be run on MTS by using:

```
$RUN SGTA:OPTSYS.O 7=(user's file for output of C and K)
      .
      .
      .
      input data cards
      .
      .
      .
$ENDFILE
```

If the data is in an MTS file, I/O device 5 can be set equal to the file name on the $RUN command. If no output is to be made to I/O device 7, device 7 should be set equal to *DUMMY* on the $RUN command. If I/O device 7 is to be set equal to *PUNCH* remember to include on appropriate card number specification (C=...) on the MTS job card. The compilation cost for the entire program under MTS is about $3.60. The run cost for the NS = 5 example using all options which follows was $ .50 .

<u>EXAMPLE INPUT AND OUTPUT:</u> The following example is a NS = 5 test problem for the control of a tanker along a straight line when subject to a white noise current disturbance normal to the ship's path. The states are heading, yaw rate, offset from the line, offset rate, and rudder angle. A single control is the rudder command. Measurements are the heading, yaw rate and offset, each contaminated by white noise. The non-dimensionalized input is as follows:

```
$SIGNON  SGTA   PRINT=TN  C=20
[redacted]
$RUN OPTSYS.O        7=*PUNCH*
 TANKER N=5 TEST PROBLEM FROM MILLERS' PAPER
 1 1 0 0 1 1 1
  5  1  3  1
       0.0E0      1.0E0      0.0E0      0.0E0      0.0E0
     6.252E0   -1.727E0      0.0E0   -6.252E0   -6.776E0
       0.0E0      0.0E0      0.0E0      1.0E0      0.0E0
     6.622E0    0.699E0      0.0E0   -0.622E0    0.846E0
       0.0E0      0.0E0      0.0E0      0.0E0     -1.0E1
     0.274E3    0.130E2    0.208E4    0.274E3    0.694E0
       0.0E0      0.0E0      0.0E0      0.0E0      1.0E1
     0.219E2
       0.0E0    6.252E0      0.0E0    0.622E0      0.0E0
     0.44E-6
       1.0E0      0.0E0      0.0E0      0.0E0      0.0E0      0.0E0
       1.0E0      0.0E0      0.0E0      0.0E0      0.0E0      0.0E0
       1.0E0      0.0E0      0.0E0
    0.240E-6   0.348E-5   2.000E-5
      0.1E-2
/*
```

The resulting output with all options selected is as follows:

```
UNIVERSITY OF MICHIGAN DEPARTMENT OF NAVAL ARCHITECTURE AND MARINE ENGINEERING
OPTSYS STATIONARY,LINEAR OPTIMAL CONTROLLER DESIGN PROGRAM
ADAPTED FROM STANFORD UNIVERSITY OPTSYS PROGRAM JUNE 1976

 TANKER N=5 TEST PROBLEM FROM MILLERS' PAPER
 ORDER OF SYSTEM =  5
 NUMBER OF CONTROLS =  1
 NUMBER OF OBSERVATIONS =  3
 NUMBER OF PROCESS NOISE SOURCES =  1


 OPEN LOOP DYNAMICS MATRIX....F

  0.0          1.000E+00   0.0          0.0          0.0
  6.252E+00   -1.727E+00   0.0         -6.252E+00   -6.776E+00
  0.0          0.0         0.0          1.000E+00    0.0
  6.220E-01    6.990E-01   0.0         -6.220E-01    8.460E-01
  0.0          0.0         0.0          0.0         -1.000E+01


 EIGENSYSTEM OF OPEN LOOP SYSTEM....

                                           RIGHT EIGENVECTORS
REAL EIGENVALUE   ( 1)........             REAL EIGENVECTOR ( 1)........

( 0.0          )+J( 0.0         )            ( 0.0       )
                                             ( 0.0       )
                                             ( 1.0000000)
                                             ( 0.0       )
                                             ( 0.0       )

REAL EIGENVALUE   ( 2)........             REAL EIGENVECTOR ( 2)........

(-2.6533882+00)+J( 0.0         )             (-0.3437864)
                                             ( 0.9121985)
                                             ( 0.0786247)
                                             (-0.2086217)
                                             ( 0.0       )

REAL EIGENVALUE   ( 3)........             REAL EIGENVECTOR ( 3)........

( 3.0438752E-01)+J( 0.0         )            ( 0.3061612)
                                             ( 0.0931916)
                                             ( 0.9063408)

                                             ( 0.2759816)
                                             ( 0.0       )

REAL EIGENVALUE   ( 4)........             REAL EIGENVECTOR ( 4)........

(-3.6390952E-15)+J( 0.0         )            ( 0.0[illegible])
                                             (-0.0[illegible])
                                             (-1.0000000)
                                             ( 0.0[illegible])
                                             ( 0.0       )

REAL EIGENVALUE   ( 5)........             REAL EIGENVECTOR ( 5)........

(-1.0000000E+01)+J( 0.0         )            (-0.0634853)
                                             ( 0.6046677)
                                             ( 0.0[illegible])
                                             (-0.1119735)
                                             ( 0.7960099)
```

MODAL MATRIX INVERSE: LEFT EIGENVECTOR MATRIX....T INVERSE

-1.852E+16	-2.116E+15	1.000E+00	2.127E+16	3.234E+15
-2.317E+00	7.529E-01	0.0	2.317E+00	-4.276E-01
2.268E+01	3.361E+00	-0.0	-2.268E+01	-4.072E+00
-1.852E+16	-2.116E+15	-0.0	2.127E+16	3.234E+15
0.0	0.0	0.0	0.0	1.272E+00

STATE WEIGHTING MATRIX....A

2.740E+02	0.0	0.0	0.0	0.0
0.0	1.300E+01	0.0	0.0	0.0
0.0	0.0	2.080E+03	0.0	0.0
0.0	0.0	0.0	2.740E+02	0.0
0.0	0.0	0.0	0.0	6.940E-01

CONTROL DISTRIBUTION MATRIX....G

0.0
0.0
0.0
0.0
1.000E+01

MODAL CONTROL DISTRIBUTION MATRIX....G PRIME

3.234E+16
-4.276E+00
-4.072E+01

3.234E+16
1.272E+01

RELATIVE CONTROLLABILITY OF NORMAL MODES....COS(THETA(I,J))

FOR MODE I WITH CONTROL J

1.136E-01
1.264E-01
1.253E-01
1.136E-01
1.000E+00

CONTROL WEIGHTING MATRIX....B

2.190E+01

EIGENSYSTEM OF OPTIMAL CLOSED LOOP SYSTEM..

COMPLEX EIGENVALUE(1)........

(-5.207049E+00)+J(2.315923E+00)

COMPLEX EIGENVECTOR(1)........

(-0.1485912)+J(-0.0903567)
(1.0000000)+J(0.0)
(0.0290061)+J(0.0228750)
(-0.2154504)+J(-0.0374324)
(0.5752734)+J(-0.4551780)

COMPLEX EIGENVALUE(2)........

(-1.723152E+00)+J(5.205664E-01)

COMPLEX EIGENVECTOR(2)........

(-0.5317973)+J(-0.1606566)
(1.0000000)+J(0.0)
(0.2444401)+J(-0.1390732)
(-0.3637383)+J(0.3149987)
(-0.1556376)+J(-0.5156879)

REAL EIGENVALUE (1)........ REAL EIGENVECTOR (1)........

(-8.157063E+00)+J (0.0)

(0.0848061)
(-0.6917690)
(-0.0166971)
(0.1361995)
(-0.7038710)

CONTROL GAINS....C

7.077064E+00 2.595249E+00 9.745623E+00 4.029601E+00 -9.669464E-01

CLOSED LOOP DYNAMICS MATRIX....F+GC

0.0	1.000000E+00	0.0	0.0	0.0
6.252000E+00	-1.727000E+00	0.0	-6.252000E+00	-6.776000E+00
0.0	0.0	0.0	1.000000E+00	0.0
6.220000E-01	6.990000E-01	0.0	-6.220000E-01	9.460000E-01
7.077064E+01	2.595249E+01	9.745623E+01	4.029601E+01	-1.966846E+01

STATE DISTURBANCE DISTRIBUTION MATRIX....GAMMA

0.0
6.252E+00
0.0
6.220E-01
0.0

MODAL STATE DISTURBANCE DISTRIBUTION MATRIX....GAMMA PRIME

-4.000E+00
6.148E+00
6.904E+00
2.000E+00
0.0

RELATIVE DISTURBABILITY OF NORMAL MODES....COS(PHI(I,J))

FOR MODE I BY DISTURBANCE J

2.236E-17
2.887E-01
3.380E-02
1.118E-17
0.0

POWER SPECTRAL DENSITY - STATE DISTURBANCE....Q

4.400E-07

MEASUREMENT SCALING MATRIX....H

1.000E+00	0.0	0.0	0.0	0.0
0.0	1.000E+00	0.0	0.0	0.0
0.0	0.0	1.000E+00	0.0	0.0

MODAL MEASUREMENT SCALING MATRIX....H PRIME

0.0	-3.438E-01	3.662E-01	3.639E-16	-6.049E-02
0.0	9.122E-01	9.319E-02	-1.222E-31	6.049E-01
1.000E+00	7.962E-02	9.063E-01	-1.009E+00	1.120E-02

```
RELATIVE OBSERVABILITY OF NORMAL MODES....COS(BETA(J,I))

     FOR MODE I FROM MEASUREMENT J

  0.0          3.438E-01   3.062E-01   3.639E-16   6.049E-02
  0.0          9.122E-01   9.319E-02   1.222E-31   6.049E-01
  1.000E+00    7.862E-02   9.063E-01   1.000E+00   1.120E-02


POWER SPECTRAL DENSITY - MEASUREMENT NOISE....R

  2.400E-07    0.0         0.0
  0.0          3.480E-06   0.0
  0.0          0.0         2.000E-05


EIGENSYSTEM OF ESTIMATE ERROR EQUATION.....


COMPLEX EIGENVALUE( 1)........                COMPLEX EIGENVECTOR( 1)........

(-2.695447E+00)+J( 1.110110E+00)                (-0.0873291)+J( 0.0131527)
                                                ( 0.8339821)+J(-0.0188352)
                                                ( 0.0074845)+J( 0.0013091)
                                                (-0.1450395)+J( 0.0095797)
                                                ( 1.0000000)+J( 0.0       )

REAL EIGENVALUE    ( 1)........                REAL EIGENVECTOR ( 1)........

(-9.502791E-02)+J( 0.0          )               ( 0.4558396)
                                                ( 0.7526974)
                                                ( 0.0550158)
                                                ( 0.4716369)
                                                ( 0.0      )

REAL EIGENVALUE    ( 2)........                REAL EIGENVECTOR ( 2)........

(-6.261854E-07)+J( 0.0          )               (-0.0057525)
                                                ( 0.0016184)
                                                ( 0.9999794)
                                                (-0.0048479)
                                                ( 0.0      )

REAL EIGENVALUE    ( 3)........                REAL EIGENVECTOR ( 3)........


(-1.000000E+01)+J( 0.0          )               ( 0.0601230)
                                                ( 0.1443075)
                                                (-0.9874144)
                                                (-0.0239471)
                                                ( 0.0      )


COVARIANCE OF THE ESTIMATION ERROR....P

 5.241248E-07   6.313576E-07   2.668216E-07   5.192288E-07   0.0
 6.313576E-07   2.980776E-06   1.090007E-07   8.463483E-07   0.0
 2.668216E-07   1.090007E-07   1.930494E-06   2.431976E-07   0.0
 5.192288E-07   8.463483E-07   2.431976E-07   5.383545E-07   0.0
 0.0            0.0            0.0            0.0            0.0


FILTER STEADY STATE GAINS....K

   2.183853E+00  1.914246E-01  1.334108E-02
   2.630657E+00  8.565454E-01  5.450034E-03
   1.111757E+00  3.132204E-02  9.652471E-02
   2.163454E+00  2.432035E-01  1.215988E-02
   0.0           0.0           0.0


COVARIANCE OF THE ESTIMATE....X HAT

 1.975776E-06  -6.313576E-07  -1.053269E-06   1.435350E-06   3.573102E-06
-6.313576E-07   1.844640E-05  -2.063580E-06  -4.146184E-06  -1.147418E-06
-1.053269E-06  -2.063590E-06   1.239787E-06  -2.431976E-07  -7.144734E-07
 1.435350E-06  -4.146184E-06  -2.431976E-07   1.923523E-06   2.997360E-06
 3.573102E-06  -1.147418E-06  -7.144734E-07   2.997360E-06   1.392963E-05
```

STATE COVARIANCE MATRIX....X = X HAT + P

2.399903E-06	2.734064E-19	-7.864673E-07	1.954579E-06	3.573102E-06
2.184022E-19	2.142717E-05	-1.954579E-06	-3.299836E-06	-1.147418E-06
-7.864673E-07	-1.954579E-06	3.170282E-06	7.142870E-20	-7.144734E-07
1.954579E-06	-3.299836E-06	4.536920E-20	2.459878E-06	2.997360E-06
3.573102E-06	-1.147418E-06	-7.144734E-07	2.997360E-06	1.332963E-05

CONTROL COVARIANCE....E(U*UT)

3.030834E-05

STATE RMS RESPONSE

1.549162E-03
4.624949E-03
1.780528E-03
1.568399E-03
3.732241E-03

CONTROL RMS RESPONSE

5.505301E-03

STEADY DISTURBANCE....W SUB C

1.000000E-03

STEADY STATE VALUES OF STATES....

-1.000E-03
-0.0
7.262E-04
-0.0
-1.626E-19

STEADY STATE CONTROLS....

-7.101E-19

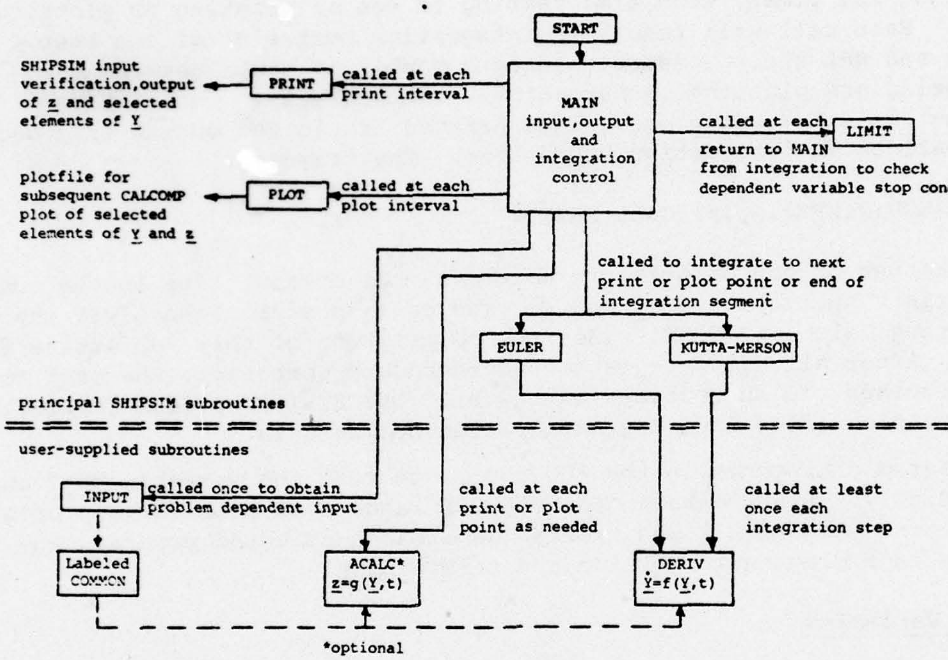
EXECUTION TERMINATED 11:14:11 T=.321 PC=0 $.31

Appendix D. Programmer's Documentation for SHIPSIM

I. Program Organization

This programmer's documentation does not duplicate the User's Documentation for SHIPSIM which is included as Appendix A. The reader should consult Appendix A before proceeding.

The macro-flow chart for SHIPSIM is shown below.



II. Main Program

The main program may be divided into the following sections: input, initialization, input verification, integration control, and error processing.

The input portion calls the user-supplied subroutine INPUT for acquisition of problem dependent data and reads those SHIPSIM integration and output control parameters given in the User's Instructions (Appendix A).

Initialization begins with the assignment of initial values to the NEQ2 elements of Y. (NEQ2 is the SHIPSIM equivalent of the INPUT subroutine variable NEQ.) In the same loop LY, the subscript of the element of Y to be tested against YTERM for the dependent variable integration stopping condition, is assigned. The value of LY is found by matching the character variable YTEST against the vector of Y-element names YLABLE. This technique is utilized frequently throughout the program. PLY and PLZ, vectors containing the subscripts of the Y and z elements to be plotted, are loaded next using a similar character matching technique.

All SHIPSIM input is written on I/O channel 6 for user verification.

Integration control begins with a call to PLOT1 for loading of the plot vector with the initial values of $\underline{Y}$. All remaining integration control operations are enclosed in the DO loop beginning at statement 4ØØ. They are executed once for each integration segment.

The call to entry PRINT1 in subroutine PRINT results in the printing of all or selected $\underline{Y}$ and all $\underline{z}$ values at the beginning of each segment. For the first segment these will be, for $\underline{Y}$, the values of $\underline{Y\emptyset}$. NST is the number of steps (STEP) in the segment, each step being the minimum of the times between printing and plotting output points. MAIN will call the appropriate integration subroutine NST times, each time testing to see if printing or plotting is required. Each call will result in integration over a Δt of one step (STEP). NPR and NPL are the nearest integer number of steps between each pair of printing and plotting output points, respectively. One of these variables must have the value one. Thus printed or plotted output will occur after each call to an integration subroutine. The statement

```
IF(MOD(J,NPR).EQ.Ø) CALL PRINT2 ...
```

determines whether or not printing is needed at the current time in the run and if so, prints the values of $\underline{Y}$ and $\underline{z}$. The call to subroutine LIMIT checks for the limiting value of YTEST. (See the description of this subroutine in section IV.) After all NIS integration segments are completed, the call to entry PLOT2 invokes the subroutines in the PLOT DESCRIPTION SYSTEM (*PLOTSYS) which generate plot files from the integration output data.

MAIN program statements in the 600-699 range read the variable NEXT and branch accordingly. Statements in the 700-999 range deal with various program error conditions. As SHIPSIM was written as a batch-oriented program, any error results in a diagnostic message and termination.

III. COMMON Variables

SHIPSIM contains the following COMMON blocks:

(unlabeled)	NEQ2
/OUTPUT/	YLABLE, ZLABLE, TITLE
/COM1/	CPRINT, IOUT, NAC
/COM2/	CPLOT, PLY, PLZ, PLN

These COMMON variables have the following definitions:

CPLOT	vector of labels of variables for which plots are desired, dimension (9) with only (PLN) used.
CPRINT	vector of labels of variables to be printed, dimension (9).
IOUT	output format switch, =1 for tabular output =2 for vectorial output
NAC	dimension of $\underline{z}$
NEQ2	dimension of $\underline{Y}$
PLN	number of plots desired, dimension of CPLOT

PLY	vector of subscripts of the elements of Y to be plotted, dimension (9).
PLZ	vector of subscripts of the elements of z to be plotted, dimension (5).
TITLE	User's 72 character title, dimension (9).
YLABLE	labels for NEQ elements of Y, dimension (25).
ZLABLE	labels for NAC elements of z, dimension (5).

All real variables above are double precision. As the library subroutines used in SHIPSIM subroutine PLOT require single precision arguments, the /COM2/ variables in that subroutine will have dimensions of two times those given above.

IV. Subroutine Descriptions

The SHIPSIM subroutines are described here in alphabetical order. The designation (I) after a subroutine calling argument indicates that the quantity is input to the subroutine; (O) indicates that the quantity is returned to the calling program.

1. NAME: DFEQKD

PURPOSE: This subroutine integrates from T to T+STEP by Kutta-Merson variable step-size integration.

CALLING SEQUENCE: MAIN, DFEQKD

ARGUMENTS:

NEQ	dimension of Y (I)
X	independent variable (I)
FIRSTP	initial integration (I) step-size
STEP	time interval through which the equations are to be integrated. (I)
Y	vector of dependent variables (I/O)
FUNCT	name of subroutine which calculates derivatives of Y. (I)
EPS	maximum permissable relative error. (I)
AB	maximum permissible absolute error. (I)
NCUTS	maximum number of times the step-size may be halved before the error return is taken. (I)
STPSZ	=.TRUE., integration progress is printed at each step. =.FALSE., printing is suppressed. (I)

SUBPROGRAMS CALLED: FUNCT

COMMENTS: An initial call to DFEQKD with EPS=Ø initializes HC, the integration step-size.

2. NAME: EULER

PURPOSE: To perform rectangular integration of Y from X to X+STEP

CALLING SEQUENCE: MAIN, EULER

ARGUMENTS:

X	independent variable (I)
Y	dependent variable vector (I/O)
STEP	time interval through which the equations are to be integrated (I)
FIRSTP	integration step-size (I)

SUBROGRAMS CALLED: TEST, DERIV

COMMENTS: Euler integration is described in Appendix A.

3. NAME: LIMIT

PURPOSE: To terminate the simulation run if Y(LY) has reached the user-specified limit YTERM

CALLING SEQUENCE: MAIN, LIMIT

ARGUMENTS:

Y	the dependent variable vector. (I)
YØ	initial values for Y. (I)
YTERM	name of the element of Y to be tested. (I)

COMMENTS: None

4. NAME: PLOT, PLOT1, PLOT2

PURPOSE: To generate linear-rectangular plots of selected elements of Y for MTS graphic post-processing.

CALLING SEQUENCE: MAIN, PLOT

ARGUMENTS:

T	independent variable. (I)
Y	dependent variable vector. (I)
NAC	dimension of z (I)
SF	scaling factor for plot size. (I)

SUBROUTINES CALLED: ACALC, PLTSIZ, PLTXMX, PSCALE, PAXIS, PGRID, PLTOFS, PLTREC, PLIN2, PSYMB, PLTEND, PXMARG

COMMENTS: This subroutine utilizes the MTS graphic subroutine library contained in the file *PLOTSYS. The subroutines in this library perform such functions as axis definition, plot scaling, labeling, and grid definition. For a complete description see MTS Vol. 11.

The library subroutines require two vectors for construction of a plot, one of values of the independent variable, and one for the dependent variable. The SHIPSIM independent variable is T, the dependent variables are selected elements of Y and z. Values of T are loaded into the vector TP through ENTRY PLOT1, one value at each call. This program segment also loads the PLNY elements of Y and PLNZ elements of z into YPLOT. MP is the current subscript for TP, LP is the subscript of YPLOT corresponding to the first element of Y and z to be loaded. To clarify, if Y(2), Y(3), and z(2) are to be plotted the variables and vectors used in PLOT would appear as:

PLY. . . . 2, 3, Ø, Ø, Ø, Ø, Ø, Ø, Ø
PLZ. . . . 2, Ø, Ø, Ø, Ø
PLN. . . . 3
PLNY . . . 2
PLNZ . . . 1

TP(MP)	LP	n	YPLOT (n) contains
MP = 1	1	1	Y(2) at T = TP(1)
		2	Y(3) at T = TP(1)
	3	3	z(1) at T = TP(1)
2	4	4	Y(2) at T = TP(2)
		5	Y(3) at T = TP(2)
	6	6	z(3) at T = TP(2)
3	7	7	Y(2) at T = TP(3)
.	.	.	.
.	.	.	.
.	.	.	.

Entry at PLOT2 occurs after a run has ended. TP(MP) contains the final value of T, while MP indicates the number of points for each plot. The *PLOTSYS subroutines are described in MTS Vol. 11 and will not be included here.

5. NAME: PRINT, PRINT1, PRINT2

PURPOSE: To print the integration results.

CALLING SEQUENCE: MAIN, PRINT

ARGUMENTS:

IS	integration segment number (I)
T	independent variable. (I)
Y	dependent variable vector (NEQx1). (I)

SUBROUTINES CALLED: ACALC

COMMENTS: The initial call to PRINT results in the loading of the vector PRY, with the subscripts of $\underline{Y}$ to be printed. This is accomplished in a manner similar to that used for loading PLY in the main program (see section II of this appendix). PRN is the number of elements of Y to be printed.

Entry at PRINT1 occurs at the beginning of each integration segment.

Entry at PRINT2 results in the printing of the current value of the independent variable T and the corresponding values of the selected elements of $\underline{Y}$ and $\underline{z}$.

V. SHIPSIM in MTS

SHIPSIM was written for the Michigan Terminal System (MTS). The program was deliberately structured so as to be as system-independent as possible to aid in its adaptation to other systems. Only two areas of system dependency exist.

Source code was written for the IBM FORTRAN IV-G (Extended) compiler. If this compiler is unavailable, program alterations must be made to suit the user's system capability.

Reference was made in sections II and IV of this appendix to the Plot Description System graphics subroutine library contained in the MTS public file *PLOTSYS. If the user's system has a similar graphics subroutine library, SHIPSIM subroutine PLOT may be adapted to suit. If it is necessary to delete the plotting capability altogether, the following changes should be made:

Line Number	Changes
10	delete
29	delete "PLN" and "SF"
35	delete
36	delete "PLD(I)"
39	change "4I5" to "3I5"
	delete "D1Ø.r"
45	delete
56-77	delete
92	delete
96	delete
116,118	delete "PLD(I)"
125	change "6" to "5"
129	delete
132,133	delete
135-138	delete
144,152	change "STE" to "PRD(I)"
146,153	delete "IF(MOD(J,NPR).EQ. Ø)"
147,154	delete
161	delete

In addition, the entire subroutine PLOT (lines 286-377) should be deleted.

Program size statistics given below apply to SHIPSIM with the plotting capability, as compiled on the Amdahl 470/V-6 computer:

Main program	221	source lines
	166	source statements
	733Ø	object bytes
Subroutine PRINT	61	source lines
	39	source statements
	2492	object bytes
Subroutine PLOT	93	source lines
	66	source statements
	14,48Ø	object bytes

Subroutine EULER	22	source lines
	17	source statements
	810	object bytes
Function TEST	11	source lines
	6	source statements
	404	object bytes
Subroutine LIMIT	17	source lines
	11	source statements
	642	object bytes
Subroutine DFEQKD	99	source lines
	97	source statements
	4080	object bytes
Total	530	source lines
	385	source statements
	30328	object bytes

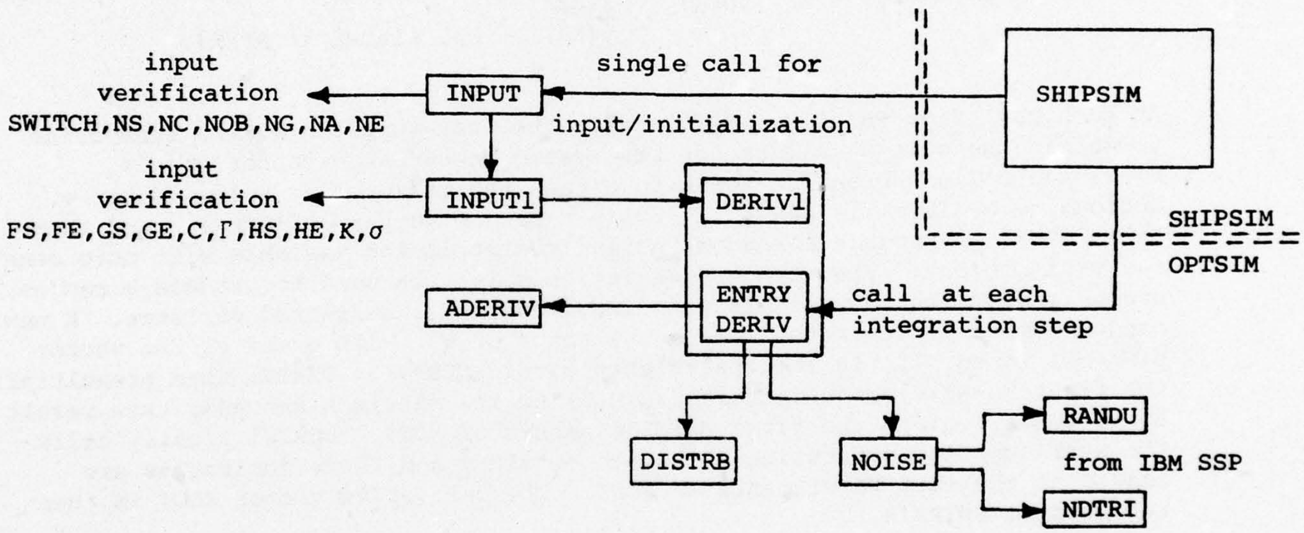
Appendix E: Programmer's Documentation for OPTSIM

I. Program Organization

This programmer's documentation does not duplicate the User's Documentation for OPTSIM which is included in Appendix B. The reader should consult both Appendix A: User's Documentation for SHIPSIM and Appendix B prior to reading this appendix.

OPTSIM is a group of eight double-precision subroutines which are run under the control of the SHIPSIM continuous systems simulation program described in Appendices A and D. These subroutines constitute the INPUT and DERIV subroutines needed by SHIPSIM. Two of the OPTSIM subroutines RANDU and NDTRI are taken from the IBM Scientific Subrouting Package available on the Michigan Terminal System (MTS) under NAAS:SSP.

A macro-flow chart for the OPTSIM subroutines running under SHIPSIM is as follows:



To obtain input to OPTSIM and perform necessary initialization, SHIPSIM calls INPUT once per integration run. Subroutine INPUT reads control and dimensioning variables and writes the input verification of these quantities. It then calls INPUT1 which reads the ten matrices and vectors which define the system, controller, and estimator (if SWITCH=1) or reads changes to these quantities (if SWITCH=2). INPUT1 writes input verification for all data or changes it reads. INPUT1 then calls DERIV1 to transmit the input data. DERIV1 initializes the integer seed XI for the random number generator and calculates the matrix A from the input data; i.e.,

$$\begin{bmatrix} \dot{\underline{x}} \\ \dot{\hat{\underline{x}}} \end{bmatrix} = \begin{bmatrix} F_s & G_sC \\ KH_s & (F_e+G_eC-KH_e) \end{bmatrix} \begin{bmatrix} \underline{x} \\ \hat{\underline{x}} \end{bmatrix} + \begin{bmatrix} \Gamma\underline{w} \\ K\underline{v} \end{bmatrix} = A \begin{bmatrix} \underline{x} \\ \hat{\underline{x}} \end{bmatrix} + \underline{b} \qquad (1)$$

This completes the problem initialization and control is then returned to SHIPSIM through INPUT1 and INPUT.

SHIPSIM calls subroutine DERIV1 at ENTRY DERIV once during each integration step throughout the simulation. This call is to obtain the derivative vector YDOT(NEQ) given current values of Y(NEQ) and TIME. The number of integrated equations NEQ equals the number of system states in $\underline{x}$(NS) plus the number of estimator states in $\underline{\hat{x}}$(NE) plus the number of additional derivatives in $\underline{y}$(NA). The correspondence between SHIPSIM and OPTSIM variables and derivatives is as follows:

$$\underline{\text{SHIPSIM}} \qquad \underline{\text{OPTSIM}}$$

$$\text{Y(NEQ)} = \text{X(NEQ)} \equiv \begin{bmatrix} \underline{x}\text{(NS)} \\ \underline{\hat{x}}\text{(NE)} \\ \underline{y}\text{(NA)} \end{bmatrix}$$

$$\text{YDOT(NEQ)} = \text{XDOT(NEQ)} \equiv \begin{bmatrix} \dot{\underline{x}}\text{(NS)} \\ \dot{\underline{\hat{x}}}\text{(NE)} \\ \dot{\underline{y}}\text{(NA)} \end{bmatrix} \begin{array}{l} \left.\begin{array}{l} \\ \\ \end{array}\right\} \text{calculated in DERIV} \\ \\ \text{calculated in ADERIV} \end{array}$$

At each call from SHIPSIM, DERIV1 calls the user-supplied DISTRB subroutine to obtain the current values for the system disturbance vector $\underline{w}$ (if NG≠0) and calls subroutine NOISE to obtain the measurement noise vector $\underline{v}$. Subroutine NOISE calls RANDU to obtain a random number $0. \leq \text{YFL} \leq 1.$ and then calls NDTRI to produce a random, normally distributed variable with zero mean and variance one. The standard deviation σ is then used to produce a random, normally distributed variable with zero mean and the desired variance. A new random number is generated for each element of $\underline{v}$. With $\underline{w}$ and $\underline{v}$, the vector $\underline{b}$(NS+NE) in eq. (1) is then calculated within DERIV1. DERIV1 then premultiplies the first NS+NE element vector of X(NEQ) by the matrix A and adds this result to vector $\underline{b}$ to form the first NS+NE elements of XDOT. DERIV1 finally calls the user-supplied subroutine ADERIV to obtain $\dot{\underline{y}}$ and these derivaties are loaded as the last NA elements of XDOT. The derivative vector XDOT is then returned to SHIPSIM.

II. COMMON Variables

The OPTSIM subroutines utilize two labeled COMMON blocks which are unique to these subroutines. These blocks appear as follows:

```
COMMON/ONE/SWITCH
COMMON/TWO/FS,FE,GS,GE,C,GAMMA,HS,HE,K,SIGMA
```

These COMMON variables have the following definitions:

- C — feedback control gains matrix; dimension (8,10) with only (NCxNE) utilized.
- FE — estimator open-loop dynamics matrix; dimension (10,10) with only (NExNE) utilized; equals FS if NE=NS.

FS — system open-loop dynamics matrix; dimension (10,10) with only (NSxNS) utilized.

GAMMA — system disturbance distribution matrix; dimension (10,10) with only (NSxNG) utilized.

GE — estimator control distribution matrix; dimension (10,8) with only (NExNC) utilized; equals GS if NE=NS.

GS — system control distribution matrix; dimension (10,8) with only (NSxNC) utilized.

HE — estimator measurement scaling matrix; dimension (10,10) with only (NOB,NE) utilized; equals HS if NE=NS.

HS — system measurement scaling matrix; dimension (10,10) with only (NOB,NS) utilized.

K — Kalman-Bucy filter gains matrix. Dimension (10,10) with only (NExNOB) utilized.

SIGMA — Vector of standard deviations for zero mean measurement noise vector $\underline{v}$(NOB). Dimension (10) with only (NOB) utilized.

SWITCH — Input control integer. If SWITCH=1 all input data is read in INPUT1. If SWITCH=2 only changes to data are read in INPUT1 using a NAMELIST read where LIST1 is defined as:

NAMELIST/LIST1/FS,FE,GS,GE,C,GAMMA,HS,HE,K,SIGMA

III. SUBROUTINE Descriptions

The OPTSIM subroutines are described here in alphabetical order. The designation (I) after a subroutine argument signifies that the quantity is input to the subroutine; (O) signifies that the quantity is output of the subroutine.

1. NAME: ADERIV

PURPOSE: This subroutine calculates up to 5 additional time derivatives YDOT=f(t,$\underline{x}$,$\underline{\hat{x}}$,$\underline{y}$) which are to be integrated as part of the simulation but which are not in $\underline{x}$ or $\underline{\hat{x}}$ in eqn. (1). This is a user-supplied subroutine as defined in the User's Documentation for OPTSIM. A dummy version without executable code is in file ADERIV.D to allow program loading without an MTS error message.

CALLING SEQUENCE: SHIPSIM, ENTRY, DERIV, ADERIV

ARGUMENTS

TIME	simulation independent variable (I)
X	NEQ vector of integrated dependent variables consisting of $\underline{x}$(NS), $\underline{\hat{x}}$(NE), and $\underline{y}$(NA). (I)
YDOT	NA vector of additional derivatives (O)
NA	dimension of YDOT. (I)
NEQ	dimension of X; NS+NE+NA (I)

SUBROUTINES CALLED:	Defined by user.
COMMENTS:	See User's Documentation for OPTSIM

2. NAME: ENTRY DERIV

PURPOSE: This entry point in SUBROUTINE DERIV1 calculates and/or loads the vector of derivatives which are integrated by SHIPSIM.

CALLING SEQUENCE: SHIPSIM, ENTRY DERIV

ARGUMENTS:

TIME	simulation independent variable. (I)
X	vector of integrated dependent variables consisting of $\underline{x}$(NS), $\hat{\underline{x}}$(NE), and $\underline{y}$(NA); this is vector Y in SHIPSIM; dimension (25) with only (NEQ) utilized. (I)
XDOT	vector of dependent variable first derivatives with respect to TIME; this is vector YDOT in SHIPSIM; dimension (25) with only (NEQ) utilized. (O)

SUBROUTINES CALLED: ADERIV, DISTRB, NOISE

COMMENTS: See Program Organization. See eqn. (1) for definition of internal matrix A (20,20) and vector B (20). Only (NS+NE, NS+NE) of A and only (NS+NE) of B are utilized in any particular run.

3. NAME: DERIV1

PURPOSE: This portion of SUBROUTINE DERIV1 obtains system, controller, and estimator input from INPUT1 via COMMON/TWO/, initializes the random number seed XI for use in RANDU, and loads the matrix A in eqn. (1).

CALLING SEQUENCE: SHIPSIM, INPUT, INPUT1, DERIV1

ARGUMENTS:

NS	dimension of state vector $\underline{x}$ (I)
NC	dimension of control vector $\underline{u}$. (I)
NOB	dimension of measurements vector $\underline{z}$. (I)
NG	dimension of system disturbance vector $\underline{w}$. (I)
NA	dimension of additional variable vector $\underline{y}$. (I)
NE	dimension of estimate vector $\hat{\underline{x}}$. (I)
NSE	NS+NE (I)
NEQ	NS+NE+NA (I)

SUBROUTINES CALLED: none

COMMENTS: see ENTRY DERIV.

4. NAME: DISTRB

PURPOSE: This subroutine calculates the system disturbance vector $\underline{w}$ using current values of simulation independent and dependent variables. This is a user-supplied subroutine as defined in the User's Documentation for OPTSIM. A dummy version without

executable code is in file DISTRB.D to allow program loading without an MTS error message.

CALLING SEQUENCE: SHIPSIM, ENTRY DERIV, DISTRB

ARGUMENTS:

TIME	simulation independent variable (I)
X	NEQ vector of integrated dependent variables consisting of $\hat{\underline{x}}$(NS), $\underline{x}$(NE), and $\underline{y}$(NA). (I)
W	NG vector of system disturbances $\underline{w}$ (O)
NG	dimension of W (I)
NEQ	dimension of X; NS+NE+NA (I)

SUBROUTINES CALLED: Defined by user.

COMMENTS: See User's Documentation for OPTSIM.

5. NAME: INPUT

PURPOSE: Reads and writes input verification for input control integer SWITCH and problem dimensions NS, NC, NOB, NG, NA, and NE.

CALLING SEQUENCE: SHIPSIM, INPUT

ARGUMENTS:

NEQ	dimension of integrated dependent variable vector; NS+NE+NA (O)
*	error return for END OF FILE on I/O device 5 or data input problem.

SUBROUTINES CALLED: INPUT1

COMMENTS: none

6. NAME: INPUT1

PURPOSE: Reads and writes input verification for matrices FS, FE, GS, GE, C, GAMMA, HS, HE, K, and SIGMA. Loads these matrices into COMMON/TWO/. If NS=NE, the subroutine reads only FS, GS, C, GAMMA, HS, K and SIGMA and sets FE=FS, GE=GS, and HE=HS.

CALLING SEQUENCE: SHIPSIM, INPUT, INPUT1

ARGUMENTS:

NS	dimension of state vector $\underline{x}$ (I)
NC	dimension of control vector $\underline{u}$. (I)
NOB	dimension of measurements vector $\underline{z}$. (I)
NG	dimension of system disturbance vector $\underline{w}$. (I)
NA	dimension of additional variable vector $\underline{y}$. (I)
NE	dimension of estimate vector $\hat{\underline{x}}$. (I)
NSE	NS+NE (I)
NEQ	NS+NE+NA (I)

SUBROUTINES CALLED: DERIV1

COMMENTS: none

7. NAME: NDTRI

PURPOSE: Returns a zero mean, normally distributed variable with variance one given $0. \leq YFL \leq 1.0$. IBM Scientific

AD-A045 444 MICHIGAN UNIV ANN ARBOR DEPT OF NAVAL ARCHITECTURE --ETC F/G 9/2
SHIPSIM/OPTSIM: SIMULATION PROGRAM FOR STATIONARY LINEAR OPTIMA--ETC(U)
JUN 77 M G PARSONS, J E GREENBLATT N00014-76-C-0751

UNCLASSIFIED 188 ONR-CR215-249-1 NL

2 OF 2
AD
A045444

END
DATE
FILMED
11-77
DDC

Subroutine Package subroutine.

CALLING SEQUENCE: SHIPSIM, ENTRY DERIV, NOISE, NDTRI

ARGUMENTS:

YFL	input variable (I)
X	output variable (O)
D	output density F(x); not used here (O)
IER	error code which equals 1 if YFL<0, or YFL>1.; not used here since RANDU will return YFL in the required range (O)

SUBROUTINES CALLED: none

COMMENTS: See IBM Scientific Subroutine Package documentation for additional details and listing.

8. NAME: NOISE

PURPOSE: Calculates random measurement noise vector with standard deviations given by SIGMA.

CALLING SEQUENCE: SHIPSIM, ENTRY DERIV, NOISE

ARGUMENTS:

SIGMA	NOB vector of measurement noise standard deviations from zero mean (I)
V	NOB vector of measurement noise (O)
NOB	dimension of measurement vector (I)
IX	integer seed for random number generator (I)

SUBROUTINES CALLED: RANDU, NDTRI

COMMENTS: Updates integer seed IX to IY after each call to RANDU.

9. NAME: RANDU

PURPOSE: Generates a random number $0. \leq YFL \leq 1.0$. IBM Scientific Subroutine Package subroutine.

CALLING SEQUENCE: SHIPSIM, ENTRY DERIV, NOISE, RANDU

ARGUMENTS:

IX	input integer seed (I)
IY	output integer seed for next call (O)
YFL	random number $0. \leq YFL \leq 1.0$ (O)

SUBROUTINES CALLED: none

COMMENTS: See IBM Scientific Subroutine documentation for additional details and listing.

Appendix F: Listing of SHIPSIM

```
C
C...     UNIVERSITY OF MICHIGAN
C...      DEPARTMENT OF NAVAL ARCHITECTURE AND MARINE ENGINEERING
C...      CONTINUOUS SYSTEMS SIMULATION PROGRAM
C
      IMPLICIT REAL*8 (A-H,O-$)
      COMMON NEQ2
      COMMON/OUTPUT/YLABLE,ZLABLE,TITLE
      COMMON/COM1/CPRINT(9),IOUT,NAC
      COMMON/COM2/CPLOT,PLY,PLZ,PLN
      INTEGER PRN,PLY,PRY,PLN,PLZ
      LOGICAL TEST
      EXTERNAL DERIV
      DIMENSION TF(5), METHOD(5), PRD(5), PLD(5), AB(5), EPS(5),
     *  NCUTS(5), Y(25), Z(5), Y0(25), FIRSTP(5), DY(25),
     *  YLABLE(25), TITLE(9), ZLABLE(5), CPLOT(9), PLY(9), PLZ(5)
      DATA T,Y,Z/31*0./, METHOD/5*0/,AB,EPS/10*0./,NCUTS/6/,NEXT/4/
      NAMELIST/DATA/TITLE, ZLABLE,Z,NAC,Y,TF,METHOD,PRD,PLD,AB,EPS,NCUTS
     *, Y0,FIRSTP,NIS,SF,YTEST,YTERM,CPRINT,CPLOT,PLN,IOUT
C
C...      FORMATTED DATA INPUT
C
  100 CONTINUE
      WRITE(6, 2000)
 2000 FORMAT('1UNIVERSITY OF MICHIGAN DEPARTMENT OF NAVAL ',
     *  'ARCHITECTURE AND MARINE ENGINEERING'/
     *  '0SHIPSIM CONTINUOUS SYSTEMS SIMULATION PROGRAM'//)
      IF (NEXT .EQ. 4) CALL INPUT(NEQ2, &999)
      READ(4,120,ERR=800)NIS,NAC,IOUT,PLN,SF,TITLE
      READ(4,121,ERR=810) (YLABLE(I), I=1, NEQ2)
      READ(4,125,ERR=820) (Y0(I),I=1,NEQ2)
      READ(4,126,ERR=830) YTEST, YTERM
      IF (NAC.GT. 0) READ(4, 121, ERR=840) (ZLABLE(I). I=1, NAC)
      IF(IOUT .EQ. 1) READ(4,122,ERR=850) CPRINT
      IF (PLN .GT. 0) READ(4,128,ERR=860) (CPLOT(I), I=1,PLN)
      READ(4,123,ERR=870) (METHOD(I),TF(I),FIRSTP(I),PRD(I),PLD(I),
     *  EPS(I),AB(I),NCUTS(I), I=1,NIS)
C
  120 FORMAT(4I5,D10.4/9A8)
  121 FORMAT(8(A8,2X))
  122 FORMAT(9A8)
  123 FORMAT(I5,D10.2,D10.6,2D5.2,2D10.6,I5)
  125 FORMAT(7D10.4)
  126 FORMAT(A8,D10.4)
  128 FORMAT(7(A8,2X))
  130 CONTINUE
C
C...     ASSIGN INITIAL VALUES
C
      LY=0
  200 DO 250 I=1,NEQ2
         Y(I)=Y0(I)
        IF(YLABLE(I) .EQ. YTEST) LY=I
  250 CONTINUE
C
C...     CONSTRUCT PLY AND PLZ VECTORS (SUBSCRIPTS OF Y AND Z TO BE PLOTTED).
C
      K=0
      DO 260 I=1,9
        PLY(I)=0
        DO 255 J=1,NEQ2
          IF (CPLOT(I) .NE. YLABLE(J)) GO TO 255
          K=K+1
          PLY(K)=J
  255   CONTINUE
  260 CONTINUE
      K=0
      DO 262 I=1,5
        PLZ(I)=0
  262 CONTINUE
      DO 270 I=1,PLN
        DO 265 J=1,NAC
          IF (CPLOT(I) .NE. ZLABLE(J)) GO TO 265
          K=K+1
          PLZ(K)=J
  265   CONTINUE
  270 CONTINUE
```

```
C
C...      INPUT VERIFICATION
C
       IF (NEXT .EQ. 2 .OR. NEXT .EQ. 5) CALL INPUT(NEQ2, &999)
       WRITE(6,2040) TITLE
 2040  FORMAT(1H-,9A8)
       WRITE(6,2001) (YLABLE(I), Y0(I),I=1,NEQ2)
       IF (NAC .EQ. 0) GO TO 203
       WRITE(6,2002)
       WRITE(6,2003) (ZLABLE(I),  I=1, NAC)
  203  CONTINUE
  205  CONTINUE
C
  210  IF (IOUT .EQ. 1 ) CALL PRINT
       IF (PLN .GT. 0) CALL PLOT
       IF (IOUT .EQ. 1) WRITE(6,2010)CPRINT
       IF (IOUT .EQ. 2) WRITE(6,2010) (YLABLE(I). I=1,NEQ2)
  215  CONTINUE
  220  IF (PLN .GT. 0) WRITE(6,2020) (CPLOT(I), I=1,PLN)
  225  CONTINUE
       IF (LY .GT. 0) WRITE(6,2030) YTEST, YTERM
C
 2001  FORMAT(1H-,3X,35H* * * VARIABLES AND INITIAL VALUES: //
     *        7(10X.4(A8.3H = ,D9.3,2X)/))
 2002  FORMAT(1H-,3X,'* * * AUXILIARY VARIABLES:'/1H )
 2003  FORMAT(10X.5(A8,2X)//)
 2010  FORMAT(1H-, 3X,31H* * * VARIABLES TO BE PRINTED: , 5X, 6(A8,2X)/
     *    (40X,6(A8,2X)))
 2020  FORMAT(1H-, 3X,31H* * * VARIABLES TO BE PLOTTED: , 5X, 6(A8,2X)/
     *(      39X, 6(A8, 2X)))
 2030  FORMAT(1H-,3X,'* * * LIMITING VALUE OF ',A8,' IS ',D10.4)
C
  300  CONTINUE
C
C...      INTEGRATION PARAMETER CHECK
C
  330  WRITE(6,3030)
       DO 332 I=1, NIS
       IF (METHOD(I) .EQ. 1) WRITE(6,3032) I, TF(I), PRD(I), PLD(I),
      * FIRSTP(I)
       IF (METHOD(I) .EQ. 2) WRITE(6.3034) I. TF(I), PRD(I), PLD(I),
      * FIRSTP(I), EPS(I)  AB(I), NCUTS(I)
  332  CONTINUE
 3030  FORMAT(1H-, 3X, 38H* * * INTEGRATION CONTROL PARAMETERS: /
     *  1H0,9X,'SEGMENT  METHOD',4X,'TF',10X,'PRD',9X,'PLD',8X,'FIRSTP',
     *  6X,'EPS',9X,'AB',7X,'NCUTS'/1H )
 3032  FORMAT(1H ,12X,I1,5X,'EULER',4(2X,D10.4))
 3034  FORMAT(1H ,12X,I1,5X,'K-M  ',6(2X,D10.4),3X,I2)
C
C...      INTEGRATION CONTROL
C
       IF(PLN .GT. 0) CALL PLOT1(T,Y0,NAC)
  400  DO 420 I=1, NIS
          CALL PRINT1(I,T,Y)
          STEP=DMIN1(PRD(I), PLD(I))
          IF (PLN .EQ. 0) STEP=PRD(I)
          NST=(TF(I)-T+STEP/2.)/STEP
          NPR=(PRD(I)+STEP/2.)/STEP
          NPL=(PLD(I)+STEP/2.)/STEP
          IF (NPR .EQ. 0) NPR=1
          IF (NPL .EQ. 0) NPL=1
  404     CONTINUE
          IF (METHOD(I) .EQ. 1)  GO TO 410
          CALL DFEQKD(0, T, FIRSTP(I), Y, DERIV, EPS(I). AB(I).
     *    NCUTS(I). &704, .FALSE.)
          DO 406 J=1,NST
             CALL DFEQKD(NEQ2, T. STEP. Y. DERIV. EPS(I). AB(I).
     *       NCUTS(I),&704,.FALSE.)
             IF (MOD(J,NPR) .EQ. 0) CALL PRINT2(T,Y)
             IF (MOD(J,NPL) .EQ. 0 .AND. PLN .GT. 0) CALL PLOT1(T,Y.NAC)
             IF (LY .GT.0 .AND. MOD(J,NPR) .EQ. 0) CALL LIMIT(Y.Y0.YTERM.LY.&460)
  406     CONTINUE
       GO TO 460
  410     DO 420 J=1,NST
             CALL EULER(T,Y, STEP, FIRSTP(I), &708)
             IF (MOD(J,NPR) .EQ. 0) CALL PRINT2(T,Y)
             IF (MOD(J,NPL) .EQ. 0 .AND. PLN .GT. 0)
     *          CALL PLOT1(T,Y.NAC)
  419        IF (LY .GT.0 .AND. MOD(J.NPR) .EQ. 0) CALL LIMIT(Y,Y0,YTERM,LY,&460)
  420  CONTINUE
  440  FORMAT(I2/9A8/(8(A8,2X)))
  450  FORMAT(5(D10.4,2X))
  460  IF (PLN .GT. 0) CALL PLOT2(SF)
C
C...      RESET VARIABLES FOR NEXT RUN
C
```

```
      500 CONTINUE
          T=0.
    C
    C...     INPUT OPTION SELECTION
    C
      600 READ(4,602,END=720,ERR=712) NEXT
      602 FORMAT(I1)
          IF (NEXT .EQ. 5) GO TO 130
          IF (NEXT .GT. 2) GO TO 100
      605 CONTINUE
      610 READ(4,DATA,ERR=716)
          GO TO 200
    C
    C...     ERROR  MESSAGES
    C
      704 WRITE (6,706)
      706 FORMAT('-*** INTEGRATION STEP SIZE HALVED MORE THAN ',
         *    'NCUTS TIMES. RUN TERMINATED.')
          GO TO 600
      708 CONTINUE
          GO TO 600
      712 WRITE(6,714)
      714 FORMAT(60H-*** DATA INPUT ERROR WHILE READING NEXT AT PROGRAM STEP
         *600 )
          GO TO 999
      716 WRITE(6,718)
      718 FORMAT(57H-*** ERROR IN UNFORMATTED DATA INPUT AT PROGRAM STEP 610
         *.  )
          GO TO 999
      720 WRITE(6,722)
      722 FORMAT(35H-*** END OF FILE ENCOUNTERED ON 4. )
          GO TO 999
      724 WRITE(6,726)
      726 FORMAT(51H-*** ILLEGAL METHOD SPECIFICATION.  RUN TERMINATED. )
          GO TO 600
      800 WRITE(6,801)
      801 FORMAT(' *** DATA INPUT ERROR ON 4 WHILE READING RECORD TYPE 1')
          GO TO 999
      810 KC=3

          GO TO 900
      820 KC=4
          GO TO 900
      830 KC=5
          GO TO 900
      840 KC=6
          GO TO 900
      850 KC=7
          GO TO 900
      860 KC=8
          GO TO 900
      870 KC=9
      900 WRITE(6,910) KC
      910 FORMAT(' *** DATA INPUT ERROR ON 4 WHILE READING RECORD TYPE',I4)
      999 CONTINUE
     1000 CONTINUE
          STOP
          END
    C
    C
          SUBROUTINE PRINT
    C
    C...     THIS SUBROUTINE PRINTS THE INTEGRATION RESULTS.
    C
          IMPLICIT REAL*8 (A-H,Q-S)
          IMPLICIT INTEGER (O-P)
          DIMENSION PRY(9),TITLE(9),Y(25),YLABLE(25),
         *   Z(5),ZLABLE(5)
          COMMON NEQ2
          COMMON/OUTPUT/YLABLE,ZLABLE,TITLE
          COMMON/COM1/CPRINT(9),IOUT,NAC
    C
    C...      COMPUTE PRN AND LOAD PRY
    C
          DO 20 I=1,9
            PRY(I)=0
            DO 10 J=1,NEQ2
              IF (CPRINT(I) .EQ. YLABLE(J)) PRY(I)=J
       10   CONTINUE
       20 CONTINUE
          DO 30 PRN=1,9
            IF (PRY(PRN) .EQ. 0) GO TO 40
       30 CONTINUE
          PRN=10
       40 PRN=PRN-1
          RETURN
      100 ENTRY PRINT1(IS,T,Y)
```

```
C
C...      PRINT INTEGRATION SEGMENT NUMBER AND LABELS FOR OUTPUT VECTOR
C
      WRITE(6,105) IS
  105 FORMAT(1H4,3X,'* * * INTEGRATION SEGMENT ',I1)
      IF (IOUT .EQ. 1 .AND. NAC .EQ. 0) WRITE (6,110) (CPRINT(I), I=1,PRN)
      IF (IOUT .EQ. 1 .AND. NAC .GT. 0) WRITE (6,110) (CPRINT(I), I=1,PRN),
     *   (ZLABLE(I), I=1,NAC)
      IF (IOUT .EQ. 2 .AND. NAC .EQ. 0) WRITE (6,115) (YLABLE(K),K=1,NEQ2)
      IF (IOUT .EQ. 2 .AND. NAC .GT. 0) WRITE (6.115) (YLABLE(K).K=1,NEQ2),
     *   (ZLABLE(L),L=1,NAC)
C
  110 FORMAT('0   TIME',6X,9(A8,4X)/1H )
  115 FORMAT('0THE OUTPUT VECTOR IS:', (T25,5(A8,2X)))
      GO TO 120
      ENTRY PRINT2(T,Y)
C
C...      PRINT VALUES FOR T, Y, AND Z
C

  120 IF (NAC .GT. 0) CALL ACALC(T,Y,Z,NAC)
      IF (IOUT .EQ. 1 .AND. NAC .EQ. 0) WRITE (6,125) T,
     *   (Y(PRY(K)),K=1,PRN)
      IF (IOUT .EQ. 1 .AND. NAC .GT. 0) WRITE (6,125) T,
     *   (Y(PRY(K)),K=1,PRN), (Z(K),K=1,NAC)
      IF (IOUT .EQ. 2 .AND. NAC .EQ. 0) WRITE (6,130) T, (Y(K),K=1,NEQ2)
      IF (IOUT .EQ. 2 .AND. NAC .GT. 0) WRITE (6,130) T,
     *   (Y(K),K=1,NEQ2), (Z(L),L=1,NAC)
C
  125 FORMAT(1H ,10(D10.4,2X))
  130 FORMAT('0T= ',D10.4,(T25,5(D10.4,2X)))
  999 RETURN
       END
C
C
      SUBROUTINE PLOT
C
C...      THIS SUBROUTINE GENERATES A PLOTFILE FOR MTS
C         GRAPHIC POST PROCESSING.
C
      IMPLICIT INTEGER(O-P)
      REAL*8 T,Y,YLABLE,CPLOT,ZLABLE,Z
      DIMENSION CPLOT(9),PLY(9),PLZ(5),TITL(18),TP(300),Y(25),YLA(2),
     *   YLAB(50),ZLAB(10),YPLOT(2700),Z(5)
      COMMON/OUTPUT/YLAB,ZLAB,TITL
      COMMON/COM2/CPLOT.PLY.PLZ.PLN
C
C...      INITIALIZE PLOT VECTOR (TP AND YPLOT) SUBSCRIPTS
C
      LP=0
      MP=0
C
C...      COUNT NUMBER OF Z ELEMENTS TO BE PLOTTED
C
      DO 125 PLNZ=1,5
         IF (PLZ(PLNZ) .EQ. 0) GO TO 130
  125 CONTINUE
      PLNZ=6
  130 PLNZ=PLNZ-1
C
C...      COMPUTE NUMBER OF Y ELEMENTS TO BE PLOTTED
C
      PLNY=PLN-PLNZ
      RETURN
      ENTRY PLOT1(T,Y,NAC)
C
C...      LOAD TP AND YPLOT VECTORS
C
      IF (PLNY .EQ.0) GO TO 140
      DO 140 K=1,PLNY
         YPLOT(LP+K)=SNGL(Y(PLY(K)))
  140 CONTINUE
      LP=LP+PLNY
      IF (PLNZ .EQ. 0) GO TO 150
      CALL ACALC(T,Y,Z,NAC)
      DO 150 K=1,PLNZ
         YPLOT(LP+K)=SNGL(Z(PLZ(K)))
  150 CONTINUE
      LP=LP+PLNZ
      MP=MP+1
      TP(MP)=SNGL(T)
      RETURN
```

```
      ENTRY PLOT2(SF)
      CALL PLTSIZ(SF)
      CALL PLTXMX(10.75)
      CALL PXMARG(0.25)
      IF (PLNY .EQ. 0) GO TO 300
      DO 200 K=1,PLNY
C...  PLOT IN LINEAR-RECTANGULAR COORDINATES
C...  DEFINE X AND Y AXES AND GRID
         CALL PSCALE(10.,0.5,XMIN,DX,TP(1),MP,1)
         CALL PSCALE(7.50,0.5,YMIN,DY,YPLOT(K),MP,PLN)
         CALL PAXIS(.75,.75,'TIME',-4,10.,0.,XMIN,DX,1.0)
         YLA(1)=YLAB(2*PLY(K)-1)
         YLA(2)=YLAB(2*PLY(K))
         CALL PAXIS(.75,.75,YLA(1),8,7.5,90.,YMIN,DY,1.0)
         CALL PGRID(.75,.75,.25,.25,40,30)
         CALL PLTOFS(XMIN,DX,YMIN,DY,.75,.75)
         CALL PLTREC
C...  DRAW CURVE
         CALL PLIN2(TP(1),YPLOT(K),MP,1,PLN,0,0,1)
C...  PRINT USER SUPPLIED TITLE
         CALL PSYMB(.75,0.0,.125,TITL(1),0.,72)
         CALL PLTEND
  200 CONTINUE
      IF (PLNZ .EQ. 0) GO TO 300
      DO 300 K=1,PLNZ
C...  PLOT IN LINEAR-RECTANGULAR COORDINATES
C...  DEFINE X AND Y AXES AND GRID
         CALL PSCALE(10.,0.5,XMIN,DX,TP(1),MP,1)
         CALL PSCALE(7.50,0.5,YMIN,DY,YPLOT(PLNY+K),
     *               MP,PLN)
         CALL PAXIS(.75,.75,'TIME',-4,10.,0.,XMIN,DX,1.0)
         YLA(1)=ZLAB(2*PLZ(K)-1)
         YLA(2)=ZLAB(2*PLZ(K))
         CALL PAXIS(.75,.75,YLA(1),8,7.5,90.,YMIN,DY,1.0)
         CALL PGRID(.75,.75,.25,.25,40,30)
         CALL PLTOFS(XMIN,DX,YMIN,DY,.75,.75)
         CALL PLTREC
C...  DRAW CURVE
         CALL PLIN2(TP(1),YPLOT(PLNY+K),MP,1,PLN,0,0,1)
C...  PRINT USER SUPPLIED TITLE
         CALL PSYMB(.75,0.0,.125,TITL(1),0.,72)
         CALL PLTEND
  300 CONTINUE
      RETURN
      END
C
C
      SUBROUTINE EULER(X,Y,STEP,FIRSTP,*)
C
C...  ...INTEGRATES Y(X) FROM X TO X+STEP USING
C...     RECTANGULAR (EULER) INTEGRATION.
C
      IMPLICIT REAL*8 (A-H,O-$)
      LOGICAL TEST
      COMMON NEQ2
      DIMENSION Y(25), YDOT(25)
      FINAL=X+STEP
   10 CONTINUE
      CALL DERIV(X, Y, YDOT)
      DO 20 I=1,NEQ2
      Y(I)=Y(I)+YDOT(I)*FIRSTP
   20 CONTINUE
      X=X+FIRSTP
      IF (TEST(X, FINAL, FIRSTP/2.)) GO TO 30
      GO TO 10
   30 CONTINUE
      RETURN
      END
C
C
      LOGICAL FUNCTION TEST(A, B,TOL)
C
C...  ...TESTS FOR EQUALITY OF DOUBLE PRECISION REAL VARIABLES
C...     WITHIN SPECIFIED TOLERANCE "TOL"
C
      REAL*8 A, B, TOL
      TEST=.FALSE.
      IF (DABS(A-B) .LT. DABS(TOL)) TEST =.TRUE.
      RETURN
      END
```

```
C
C
      SUBROUTINE LIMIT(Y,Y0,YTERM,LY,*)
C     ...THIS ROUTINE TERMINATES THE RUN IF THE VALUE OF
C        A SPECIFIED ELEMENT OF Y CROSSES YTERM.
C           YTERM=LIMITING VALUE OF Y(LY)
C
      IMPLICIT REAL*8 (A-H,O-$)
      DIMENSION Y(25),YLABLE(25),Y0(25),ZLABLE(5),TITLE(9)
      COMMON/OUTPUT/YLABLE,ZLABLE,TITLE
      IF (Y(LY) .GE. YTERM .AND. Y0(LY) .LE. YTERM) GO TO 100
      IF (Y(LY) .LE. YTERM .AND. Y0(LY) .GE. YTERM) GO TO 100
      RETURN
  100 WRITE(6,200) YLABLE(LY),YTERM
  200 FORMAT('-*** VALUE OF ',A8,' HAS REACHED THE LIMITING',
     *  ' VALUE OF ',D15.6/' *** RUN TERMINATED. ')
      RETURN 1
      END
C
C
      SUBROUTINE DFEQKD(NEQ,X,STEP,Y,F,EPS,AB,NCUTS,*,STPSZ)
      IMPLICIT REAL*8 (A-H,O-$)
          INTEGER NEQ, NCUTS
      REAL*8 X,STEP,Y(3),F,EPS,AB,YY(3)
          EXTERNAL  F
          LOGICAL STPSZ
      REAL*8 HC/0.0D0/,FINAL,H2,H3,H6,H8,ERR,TEST,T,H,EPSL,TEMP
          REAL*8 Y1(30),Y2(30),F0(30),F1(30),F2(30)
          INTEGER I , CUT
          LOGICAL DBL
   50 FORMAT (' THE STEPSIZE IS NOW',1PD15.6, ' AT TAU =',D15.6)
   60 FORMAT ('    THE STEPSIZE HAS BEEN HALVED ',I3,' TIMES')
          IF(NEQ.NE.0)  GO TO 10
               HC = STEP
               RETURN
   10     IF(STEP.EQ.0)  RETURN
          IF(HC.EQ.0) HC = STEP
          FINAL = X+STEP
          H = STEP
          EPSL = EPS
          IF(EPS.EQ.0 .OR.DABS(H) .LE.DABS(HC)) GO TO 15
               IF(H*HC.LE.0)  HC = -HC
               H = HC
   15     T = X+H
          CUT = NCUTS
          X = FINAL
          H2 = H/2.
          H3 = H/3.
          H6 = H/6.
          H8 = H/8.
   20     IF (H.GT.0 .AND. T.GT.FINAL .OR. H.LT.0 .AND. T.LT.FINAL)
     C    GO TO 40
   21          CALL F(T-H,Y,F0,H,T-H,Y)
               DO 22 I = 1,NEQ
   22               Y1(I) = F0(I)*H3+Y(I)
               CALL F(T-2.*H3,Y1,F1,H,T-H,Y)
               DO 23 I = 1,NEQ
   23               Y1(I) = (F0(I)+F1(I))*H6+Y(I)
               CALL F(T-2.*H3,Y1,F1,H,T-H,Y)
               DO 24 I = 1,NEQ
   24               Y1(I) = (F1(I)*3.+F0(I))*H8+Y(I)
               CALL F(T-H2,Y1,F2,H,T-H,Y)
               DO 25 I = 1,NEQ
   25               Y1(I) = (F2(I)*4.-F1(I)*3.+F0(I))*H2 +Y(I)
               CALL F(T,Y1,F1,H,T-H,Y)
               DO 26 I = 1,NEQ
   26               Y2(I) = (F2(I)*4.+F1(I)+F0(I))*H6 +Y(I)
          IF(EPSL.EQ.0) GO TO 38
               DBL = .TRUE.
               DO 35 I = 1,NEQ
                    ERR =DABS(Y1(I)-Y2(I))*0.2
                    TEST =DABS(Y1(I))*EPSL
                    IF(ERR.LE.TEST .OR.  ERR.LT.AB) GO TO 34
                         H = H2
                         T = T-H2
                         IF (.NOT.STPSZ) GO TO 30
                              TEMP = T-H2
                              WRITE (6,50) H, TEMP
   30                    CUT = CUT - 1
                         IF (CUT .GE. 0) GO TO 31
                              X = T - H2
                              WRITE (6,60) NCUTS
                              RETURN 1
   31                    IF(T+H.NE.T) GO TO 33
                              X = T
                              RETURN 1
```

```
  33                   H2 = H/2.
                       H3 = H/3.
                       H6= H/6.
                       H8 = H/8.
                       GO TO 21
  34               IF(64.0*ERR.GT.TEST)  DBL = .FALSE.
  35           CONTINUE
               IF(.NOT.DBL) GO TO 38
                       H2 = H
                       H = 2.*H
                       IF (STPSZ)      WRITE (6,50) H,T
                       H3 = H/3.
                       H6 = H/6.
                       H8 = H/8.
                       CUT = NCUTS
  38       DO 39 I = 1,NEQ
               YY(I)=Y2(I)
  39           Y(I) = Y2(I)
           T = T+H
           GO TO 20
  40       IF(EPSL.EQ.0)  RETURN
           HC = H
           H = FINAL-(T-H)
           IF(DABS(H).LE.DABS(FINAL)*9.536744D-7)  RETURN

               T= FINAL
               EPSL = 0
               H2 = H/2.
               H3 = H/3.
               H6 = H/6.
               H8 = H/8.
               GO TO 20
        END
```

Appendix G: Listing of OPTSIM Subroutines

The two IBM Scientific Subroutine Package subroutines RANDU and NDTRI are not included here. See IBM documentation for source code listings for these subroutines. Requirements for the two user-supplied subroutines are included in the User's Documentation for OPTSIM. The source code listing of the four remaining OPTSIM subroutines are included here.

```
      SUBROUTINE INPUT(NEQ,*)
      IMPLICIT REAL*8 (A-H,O-$)
      REAL*8 K
      INTEGER SWITCH
      COMMON/ONE/SWITCH
      COMMON/TWO/FS,FE,GS,GE,C,GAMMA,HS,HE,K,SIGMA
      DIMENSION FS(10,10),FE(10,10),GS(10,8),GE(10,8),C(8,10
     *          GAMMA(10,10),K(10,10),HS(10,10),HE(10,10),SI
      READ(5,100,END=500) SWITCH
      IF (SWITCH.EQ.2) GO TO 10
      READ(5,110,END=500) NS,NC,NOB,NG,NA,NE
      NEQ=NS+NE+NA
      WRITE(6,201)
      WRITE(6,210) NEQ
      WRITE(6,220) NS
      WRITE(6,230) NC
      WRITE(6,240) NOB
      WRITE(6,250) NG
      WRITE(6,260) NA
      WRITE(6,270) NE
      NSE=NS+NE
10    CALL INPUT1(NS,NC,NOB,NG,NA,NE,NSE,NEQ)
      RETURN
500   WRITE(6,280)
      RETURN1
100   FORMAT(I3)
110   FORMAT(6I3)
20[illegible] FORMAT('0OPTSIM OPTIMAL STOCHASTIC CONTROLLER SIMULATION
[illegible] FORMAT('0INPUT VERIFICATION    NEQ ='.I3)
[illegible] FORMAT('0ORDER OF SYSTEM =',I3)
[illegible] FORMAT('0NUMBER OF CONTROLS =',I3)
[illegible] FORMAT('0NUMBER OF OBSERVATIONS =',I3)
[illegible] FORMAT('0NUMBER OF PROCESS NOISE SOURCES =',I3)
260   FORMAT('0NUMBER OF AUXILIARY STATES ='.I3)
270   FORMAT('0ORDER OF ESTIMATOR =',I3)
280   FORMAT('0END OF FILE OR DATA ERROR')
      END
```

```
      SUBROUTINE INPUT1(NS,NC,NOB,NG,NA,NE,NSE,NEQ)
      IMPLICIT REAL*8 (A-H,O-$)
      REAL*8 K
      INTEGER SWITCH
      COMMON/ONE/SWITCH
      COMMON/TWO/FS,FE,GS,GE,C,GAMMA,HS,HE,K,SIGMA
      DIMENSION FS(10,10),FE(10,10),GS(10,8),GE(10,8),C(8,10),
     *          GAMMA(10,10),HS(10,10),HE(10,10),K(10,10),SIGMA(10)
      NAMELIST/LIST1/FS,FE,GS,GE,C,GAMMA,HS,HE,K,SIGMA
      GO TO (10,160),SWITCH
      WRITE(6,200)
      CALL EXIT
10    DO 20 I=1,NS
         READ (5,100) (FS(I,J),J=1,NS)
20    CONTINUE
      WRITE(6,201)
      DO 25 I=1,NS
         WRITE(6,101) (FS(I,J),J=1,NS)
25    CONTINUE
      IF (NS.EQ.NE) GO TO 40
      DO 30 I=1,NE
         READ (5,100) (FE(I,J),J=1,NE)
30    CONTINUE
      WRITE(6,202)
      DO 35 I=1,NE
         WRITE(6,101) (FE(I,J),J=1,NE)
35    CONTINUE
      GO TO 70
```

```
40   CONTINUE
     DO 60 I=1,NS
        DO 50 J=1,NS
           FE(I,J)=FS(I,J)
50      CONTINUE
60   CONTINUE
70   CONTINUE
     READ (5,100) ((GS(I,J),J=1,NC),I=1,NS)
     WRITE(6,203)
     DO 73 I=1,NS
        WRITE(6,101) (GS(I,J),J=1,NC)
73   CONTINUE
     IF (NS.EQ.NE) GO TO 80
     READ (5,100) ((GE(I,J),J=1,NC),I=1,NE)
     WRITE(6,204)
     DO 76 I=1,NE
        WRITE(6,101) (GE(I,J),J=1,NC)
76   CONTINUE
     GO TO 110
80   CONTINUE
     DO 104 I=1, NE
        DO 90 J=1,NC
           GE(I,J)=GS(I,J)
90      CONTINUE
104  CONTINUE
110  CONTINUE
     READ (5,100) ((C(I,J),J=1,NE),I=1,NC)
     IF (NS.NE.NE)  WRITE(6,205)
     IF (NS.EQ.NE)  WRITE(6,206)
     DO 112 I=1,NC
        WRITE(6,101) (C(I,J),J=1,NE)
112  CONTINUE
     IF (NG.EQ.0) GO TO 115
     READ (5,100) ((GAMMA(I,J),J=1,NG),I=1,NS)
     WRITE(6,207)
     DO 114 I=1,NS
        WRITE(6,101) (GAMMA(I,J),J=1,NG)
114  CONTINUE
115  CONTINUE
     READ (5,100) ((HS(I,J),J=1,NS),I=1,NOB)
     WRITE(6,208)
     DO 116 I=1,NOB
        WRITE(6,101) (HS(I,J),J=1,NS)
116  CONTINUE
     IF (NS.EQ.NE) GO TO 120
     READ (5,100) ((HE(I,J),J=1,NE),I=1,NOB)
     WRITE(6,209)
     DO 118 I=1,NOB
        WRITE(6,101) (HE(I,J),J=1,NE)
118  CONTINUE
     GO TO 150
120  CONTINUE
     DO 140 I=1,NOB
        DO 130 J=1,NS
           HE(I,J)=HS(I,J)
130     CONTINUE
140  CONTINUE
150  CONTINUE
     READ (5,100) ((K(I,J),J=1,NOB),I=1,NE)
     IF (NS.NE.NE)  WRITE(6,210)
     IF (NS.EQ.NE)  WRITE(6,211)
     DO 153 I=1,NE
        WRITE(6,101) (K(I,J),J=1,NOB)
153  CONTINUE
     READ (5,100) (SIGMA(I),I=1,NOB)
     WRITE(6,212)
     DO 156 I=1,NOB
        WRITE(6,101) SIGMA(I)
156  CONTINUE
     WRITE(6,102)
     GO TO 170
160  READ (5,LIST1)
     WRITE(6,LIST1)
     WRITE(6,102)
170  CALL DERIV1(NS,NC,NOB,NG,NA,NE,NSE,NEQ)
     RETURN
100  FORMAT(6E12.5)
101  FORMAT(8E15.5)
102  FORMAT('0')
200  FORMAT('0INPUT DATA ERROR: CHECK SWITCH')
201  FORMAT('0SYSTEM OPEN LOOP DYNAMICS MATRIX....FS(NS,NS)',//)
202  FORMAT('0ESTIMATOR OPEN LOOP DYNAMICS MATRIX....FE(NE,NE)',//)
203  FORMAT('0SYSTEM CONTROL DISTRIBUTION MATRIX....GS(NS,NC)',//)
204  FORMAT('0ESTIMATOR CONTROL DISTRIBUTION MATRIX....GE(NE,NC)',//)
205  FORMAT('0FEEDBACK CONTROL GAINS....C(NC,NE)',//)
206  FORMAT('0FEEDBACK CONTROL GAINS....C(NC,NS)',//)
207  FORMAT('0SYSTEM DISTURBANCE DISTRIBUTION MATRIX....GAMMA(NS,NG)',
    *//)
208  FORMAT('0SYSTEM MEASUREMENT SCALING MATRIX....HS(NOB,NS)',//)
209  FORMAT('0ESTIMATOR MEASUREMENT SCALING MATRIX....HE(NOB,NE)',//)
210  FORMAT('0KALMAN-BUCY FILTER GAINS....K(NE,NOB)',//)
211  FORMAT('0KALMAN-BUCY FILTER GAINS....K(NS,NOB)',//)
212  FORMAT('0MEASUREMENT NOISE STANDARD DEVIATIONS....SIGMA(NOB)',//)
     END
```

```
      SUBROUTINE DERIV1(NS,NC,NOB,NG,NA,NE,NSE,NEQ)
      IMPLICIT REAL*8 (A-H,O-$)
      REAL*8 K
      COMMON/TWO/FS,FE,GS,GE,C,GAMMA,HS,HE,K,SIGMA
      DIMENSION FS(10,10),FE(10,10),GS(10,8),GE(10,8),C(8,10),
     *          GAMMA(10,10),HS(10,10),HE(10,10),K(10,10),SIGMA(10),
     *          W(10),V(10),A(20,20),B(20),X(25),XDOT(25),YDOT(5)
      IX = 1001
C *****  PUT FS IN PARTITION 1 OF A       *****
      DO 20 I=1,NS
         DO 10 J=1,NS
            A(I,J)=FS(I,J)
10       CONTINUE
20    CONTINUE
C *****  PUT +GS*C IN PARTITION 2 OF A     *****
      DO 50 I=1,NS
         DO 40 J=1,NE
            SUM=0.
            DO 30 JN=1,NC
               SUM=SUM+GS(I,JN)*C(JN,J)
30          CONTINUE
            A(I,J+NS)=SUM
40       CONTINUE
50    CONTINUE
C *****  PUT K*HS IN PARTITION 3 OF A     *****
      DO 80 I=1,NE
         DO 70 J=1,NS
            SUM=0.
            DO 60 IP=1,NOB
               SUM=SUM+K(I,IP)*HS(IP,J)
60          CONTINUE
            A(I+NS,J)=SUM
70       CONTINUE
80    CONTINUE
C *****  PUT FE IN PARTITION 4 OF A       *****
      DO 100 I=1,NE
         DO 90 J=1,NE
            A(I+NS,J+NS)=FE(I,J)
90       CONTINUE
100   CONTINUE
C *****  PUT +GE*C IN PARTITION 4 OF A     *****
      DO 130 I=1,NE
         DO 120 J=1,NE
            SUM=0.
            DO 110 JN=1,NC
               SUM=SUM+GE(I,JN)*C(JN,J)
110         CONTINUE
            A(I+NS,J+NS)=A(I+NS,J+NS)+SUM
120      CONTINUE
130   CONTINUE
C *****  PUT -K*HE IN PARTITION 4 OF A     *****
      DO 160 I=1,NE
         DO 150 J=1,NE
            SUM=0.
            DO 140 IP=1,NOB
               SUM=SUM+K(I,IP)*HE(IP,J)
140         CONTINUE
            A(I+NS,J+NS)=A(I+NS,J+NS)-SUM
150      CONTINUE
160   CONTINUE
      RETURN
      ENTRY DERIV(TIME,X,XDOT)
C  *****  CALCULATE B VECTOR    ***************************************
      DO 170 I=1,NSE
         B(I)=0.
170   CONTINUE
      IF (NG.EQ.0) GO TO 190
      CALL DISTRB(TIME,X,W,NG,NEQ)
      DO 190 I=1,NS
         DO 180 JQ=1,NG
            B(I)=B(I)+GAMMA(I,JQ)*W(JQ)
180      CONTINUE
190   CONTINUE
      CALL NOISE(SIGMA,V,NOB,IX)
      DO 210 I=1,NE
         DO 200 JP=1,NOB
            B(I+NS)=B(I+NS)+K(I,JP)*V(JP)
200      CONTINUE
210   CONTINUE
C     *****      CALCULATE XDOT       ***************************************
      DO 230 I=1,NSE
         XDOT(I)=0.
         DO 220 J=1,NSE
            XDOT(I)=XDOT(I)+A(I,J)*X(J)
220      CONTINUE
         XDOT(I)=XDOT(I)+B(I)
230   CONTINUE
      IF (NA.EQ.0) GO TO 235
      CALL ADERIV(TIME,X,YDOT,NA,NEQ)
235   CONTINUE
      DO 240 I=1,NA
         XDOT(I+NSE)=YDOT(I)
240   CONTINUE
      RETURN
      END
```

```
      SUBROUTINE NOISE(SIGMA,V,NOB,IX)
C ****** NOISE  GENERATOR    *************************************
C      SUBROUTINES RANDU AND NDTRI FROM IBM SSP
      DIMENSION SIGMA(NOB),V(NOB)
      DO 200 I=1,NOB
      CALL RANDU(IX,IY,YFL)
      CALL NDTRI (YFL,X,D,IER)
      V(I) = X*SIGMA(I)
      IX = IY
200   CONTINUE
      RETURN
      END
```

DISTRIBUTION LIST

Office of Naval Research Department of the Navy Arlington, VA. 22217 Attn: Mr. D. S. Siegel, Code 211 Dr. S. Brodsky, Code 432	 4 1
Director Office of Naval Research Branch Office 536 South Clark Street Chicago, IL. 60605 Attn: Mr. M. A. Chaszeyka	 1
Commanding Officer Naval Ship Research and Development Center Bethesda, MD. 20034 Attn: W. E. Smith, Code 1576	 1
Defense Documentation Center Cameron Station Alexandria, VA. 22134	 12
The Analytic Sciences Corp. 6 Jacob Way Reading, MA. 01867 Attn: Mr. C. Price	 1
Systems Control Inc. 1801 Page Mill Road Palo Alto, CA. 94304 Attn: Dr. E. Hall	 1
Commander Naval Sea Systems Command Washington, DC. 20362 Attn: Mr. R. P. Dilts NAVSEA 0323	 1
Commanding Officer Naval Ship Engineering Center Washington, DC. 20362 Attn: NSEC 6165	 1
Prof. A. E. Bryson, Jr. Chairman, Dept. of Aeronautics and Astronautics Stanford University Stanford, CA. 94305	 1
Prof. J. V. Cornacchio School of Advanced Technology State Univ. of New York at Binghamton Binghamton, NY. 13901	 1

CDR. Bruce C. Skinner
Chief, Marine and Ocean Engineering Section
U.S. Coast Guard Academy
New London, CT. 06320 1

Dr. Rameswar Bhattacharyya
Director of the Naval Architecture Program
U.S. Naval Academy
Annapolis, MD. 1

Dr. Haruzo Eda
Davidson Laboratory
Stevens Institute of Technology
711 Hudson Street
Hoboken, NJ. 07030 1

Dr. Masataka Fujino
Department of Naval Architecture
University of Tokyo
3-1, Hongo 7-chrome
Bunkyo-ku, Tokyo, JAPAN 1

Dr. Horst Nowacki
Technishe Universität Berlin
Salzufer 17/19
1 Berlin 10,
FEDERAL GERMAN REPUBLIC 1

Dr. J. P. Hooft
Netherlands Ship Model Basin
P.O. Box 28
6700AA Wageningen, THE NETHERLANDS 1

Mathematical Hydraulics Division
Waterways Experiment Station
PO Box 631
Vicksburg, MS. 39180
Attn: Mr. Thomas D. Ankeny 1

Prof. Dr.-Ing. H. Schneekluth
Technishe Hochschule Aachen
Templergraben 55
51 Aachen, FEDERAL GERMAN REPUBLIC 1

Office of Research and Development
Maritime Administration
Department of Commerce
Washington, DC. 20230 1

Prof. Tor Vinje
Institutt for Skipshydrodynamik
N-7034-Trondheim NTH
NORWAY 1

Prof. J.R. Pauling
Dept. of Naval Architecture
University of California
Berkeley, CA. 94720 1

Dr. Charles C. Bates, G-DS/62
U.S. Coast Guard Headquarters
400 Seventh St., S.W.
Washington, DC. 1

Prof. William F. Powers
Department of Aerospace Engr.
University of Michigan
Ann Arbor, MI. 48109 1

Dr. John Ware
Operations Research, Inc.
1400 Spring Street
Silver Spring, MD. 20910 1

American Bureau of Shipping
45 Broad Street
New York, NY. 10004
Attn: Mr. Stanley Stiansen 1